```
# VexCode-2020
Codebase for the Vex 2020-2021 Change Up season
## Included Projects
Autonomous Period Simulator

Code Generator

Macros for Documentation

PID Debugger

Code Printing

Robot Code

Automated Scouting

## License
[gpl-3.0](https://opensource.org/licenses/lgpl-3.0.html)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Jun  7 10:22:26 2019

@author: aiden
"""

import os
import treelib


class DirectoryTree:
    """
    contains methods for making a treelib tree structure of a
    given directory tree
    meant to be inherited but can be a stand alone class

    @params = None
    @return = None
    """

    def __init__(self):
        self.__directories = [] #will contain all directories and subdirectories
        self.__directory_tree = treelib.Tree() #will contain all directories ordered according
                                #to "config.txt"



    def __get_dirs(self, parent_directory, storage_list):
        """
        recursively prints all files in parent directory
        put in list given

        @params = directory to start at, directory to write output to
        @return = None
        """

        dirs = [f.path for f in os.scandir(parent_directory) if f.is_dir()]

        for directory in dirs:
            storage_list.append(directory)
            self.__get_dirs(directory, storage_list)



    def __get_all(self, parent_directory, storage_list):
        """
        recursively prints all files in parent directory
        put in list given

        @params = directory to start at, directory to write output to
        @return = None
        """

        dirs = [f.path for f in os.scandir(parent_directory) if f.is_dir()]
        files = [f.path for f in os.scandir(parent_directory)]

        for file in files:
            storage_list.append(file)

        for directory in dirs:
            storage_list.append(directory)
            self.__get_all(directory, storage_list)



    def __make_tree(self):
        """
        add directories found to a treelib structure
        the parent directory is set in "config.txt"

        @params = None
        @return = None
        """
        depth = self.__directory_tree.depth()

        #splits path of directory and makes nodes of partial paths until
        #it reaches the end of the path given
        #the resulting tree is saved in "self.__directory_tree"
        for path in self.__directories:
            path_split = path.split("/")

            i = depth

            while i < len(path_split):
                try:
                    if i > 0:
                        name = path_split[i]
                        node_id = "/".join(path_split[0:i + 1])
                        parent = "/".join(path_split[0:i])
                    else:
                        name = path_split[i]
                        node_id = path_split[i]
                        parent = None

                    self.__directory_tree.create_node(tag=name, identifier=node_id, parent=parent)

                except treelib.exceptions.NodeIDAbsentError:
                    pass
                except treelib.exceptions.DuplicatedNodeIdError:
                    pass

                i = i + 1



    def return_directory_tree(self, parent_directory):
        """
        makes and returns a treelib tree structure of a directory
        tree based on a parent directory given only directories are
        included
```

```
114:
115:        @params = string of parent directory to start node at
116:        @return = treelib tree structure of directory tree,
117:                  list of directories found
118:        """
119:        self.__directory_tree = treelib.Tree()
120:        self.__directories = []
121:
122:        self.__get_dirs(parent_directory, self.__directories)
123:        self.__make_tree()
124:
125:        return self.__directory_tree, self.__directories
126:
127:
128:
129:    def return_tree(self, parent_directory):
130:        """
131:        makes and returns a treelib tree structure of a directory
132:        tree based on a parent directory given
133:        everything is included
134:
135:        @params = string of parent directory to start node at
136:        @return = treelib tree structure of directory tree,
137:                  list of directories found
138:        """
139:        self.__directory_tree = treelib.Tree()
140:        self.__directories = []
141:
142:        self.__get_all(parent_directory, self.__directories)
143:        self.__make_tree()
144:
145:        return self.__directory_tree, self.__directories
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jun  5 19:13:03 2019

@author: aiden
"""

import math
import os
import shutil


class PostScript:
    """
    class for converting code to pictures
    """
    def __init__(self):
        self.alldirs = [".."] #list declaration that will contain all
                              #directories. Parent directory is included in it
                              #because there is code to be printed in it
                              #
                              #list will contain all directories that contain
                              #files to be printed

        self.allfiles = [] #list declaration that will contain all files that
                           #must be converted to postscript

        self.pictures_dir = "../Pictures/" #parent directory of where pictures
                                           #will be placed

        self.pictures = [] #contains locations of pictures

        self.directory_exceptions = [ #directories that contain no files to
                                      #be printed
            "../AutonSimulator/__pycache__",
            "../CodeGenerator/__pycache__",
            "../Pictures",
            "../PIDDebugging/__pycache__",
            "../Print/__pycache__",
            "../Prototypes",
            "../RobotCode/bin",
            "../RobotCode/firmware",
            "../RobotCode/include/display",
            "../RobotCode/include/okapi",
            "../RobotCode/include/pros",
            "../RobotCode/lib",
            "../RobotCode/.d",
            "../RobotCode/src/JSONLibrary",
            "../RobotCode/src/objects/lcdCode/fonts",
            "../Scouting/__pycache__",
            "../Cascades",
            "../.git"
            ]

        self.file_exceptions = [  #files in included directories of acceptable
                                  #extensions that should not be converted to
                                  #postscript to be printed
            "../license.html",
            "../PIDDebugging/ut.txt",
            "../PIDDebugging/test.txt",
            "../Print/tree.txt",
            "../Print/tree.rtf",
            "../RobotCode/include/main.h",
            "../RobotCode/include/api.h",
            "../RobotCode/compile_commands.json",
            "../RobotCode/src/objects/lcdCode/DriverControl/logo.c"
            ]

        self.valid_extensions = [ #files with this extension will be allowed to
                                  #be printed
            ".py",
            ".c",
            ".cpp",
            ".hpp",
            ".h",
            ".sh",
            ".txt",
            ".json",
            ".md"
            ]

        if not os.path.isdir(self.pictures_dir):
            os.mkdir(self.pictures_dir)  #makes directory for pictures


    def __get_directories_recursively(self, parent_directory, storage_list):
        """
        recursively prints all files in parent directory
        put in list "alldirs"

        @params = directory to start at, directory to write output to
        @return = None
        """

        dirs = [f.path for f in os.scandir(parent_directory) if f.is_dir()]

        for directory in dirs:
            storage_list.append(directory)
            self.__get_directories_recursively(directory, storage_list)



    def __find_files(self, directory):
        """
        finds all files that are not subdirectories in a given directory that
        are of a specific type

        @params = directory to look through
        @return = type list of files that are in the directory that are
                not sub directories
        """

```

```
        files = [] #contains all files that will be returned

        contents = [f.path for f in os.scandir(directory) if not f.is_dir()]
            #returns everything in directory including
            #sub directories

        for file in contents: #checks to make sure that only files that are not
                        #directories and have a valid extension are added
                        #to the list of files
            _, extension = os.path.splitext(file)
            if extension in self.valid_extensions:
                files.append(file)

        return files




    def __convert_to_postscript(self, file):
        """
        converts the specified file to postscript format which is a printable
        code type. The files are placed inside of "../pictures/" in the same
        directories that they were originally in

        @params = relative path of file to convert
        @return = None
        """

        _, extension = os.path.splitext(file)
        output_name = ((self.pictures_dir + file.split(extension)[0]
                    + extension.upper().split(".")[1]).replace("/../", "/")
                    + ".ps")

        self.pictures.append(output_name)

        #get the highlight color that will be used by shell command
        #based on the type of file it is
        if extension == ".py":
            highlight_color = "--highlight=python"

        elif extension in [".cpp", ".hpp"]:
            highlight_color = "--highlight=cpp"

        elif extension in [".c", ".h"]:
            highlight_color = "--highlight=c"

        elif extension == ".sh":
            highlight_color = "--highlight=bash"

        else:
            highlight_color = "--highlight=mail"

        #runs command that will convert a file to postscript
        #the output file location is in self.pictures
        os.system("enscript -G --line-numbers -o "
                + output_name + " "
                + highlight_color
                + " --color=1 -f Palatino-Roman5 --columns 1 "
                + file)


    def prepare_code(self):
        """
        prepares code to be printed by converting all code to
        postscript

        @params = None
        @return = None
        """
        self.allfiles = [] #list declaration that will contain all files that
                        #must be converted to postscript

        self.pictures_dir = "../Pictures/" #parent directory of where pictures
                                #will be placed

        self.pictures = [] #contains locations of pictures
        shutil.rmtree(self.pictures_dir)  #clean out pictures folder so that
                                #nothing is there that shouldn't be
        os.mkdir(self.pictures_dir)

        self.__get_directories_recursively("..", self.alldirs)
                        #gets unrefined list of directories
                        #saved in list "alldirs"
        to_remove = []
        for directory in self.alldirs: #iterates through each directory
                                #found
            for exception in self.directory_exceptions: #checks to see if
                                    #directory should
                                    #be excluded
                if exception in directory:
                    to_remove.append(directory)  #do not remove items from list
                                        #because changing the size
                                        #while iterating through the
                                        #list will cause
                                        #unexpected results

                    break #end looking for exception because if one has
                        #been found it is already in the to remove list
                        #and no other cases will match it

        alldirs = set(self.alldirs) #convert each list to type set
                        #and find the difference between them
                        #the difference will always be the
                        #directories wanted because all the items
                        #in "to_remove" will be in "alldirs" and all
                        #that is left is the directories to keep

        to_remove = set(to_remove)

        alldirs = list(alldirs - to_remove) #find difference of the two
                                #sets and convert back to type
                                #list remaining list will be
```

```
227:                                #the directories that contain
228:                                #code to be printed
229:
230:
231:
232:
233:
234:        for directory in alldirs:
235:            self.allfiles.append(self.__find_files(directory))
236:                                    #by adding
237:                                    #the list
238:                                    #directly to the all
239:                                    #files list, list will
240:                                    #be segmented and can
241:                                    #be printed in order
242:                                    #easier
243:
244:        #make directory under pictures directory for a picture of the code
245:        #to go
246:        if not os.path.isdir((self.pictures_dir + directory).replace("/../", "/")):
247:            os.makedirs((self.pictures_dir + directory).replace("/../", "/"))
248:
249:        for fileset in self.allfiles:
250:            for file in fileset:
251:                if not file in self.file_exceptions:
252:                    self.__convert_to_postscript(file)
253:
254:
255:
256:    def get_page_count(self, files=None):
257:        """
258:        gets total size of all post script files in
259:        "Pictures" directory that was made on config file
260:        in "Pictures" directory
261:
262:        @params = (optional) type list of paths of files to print
263:        @return = int amount of pages
264:        """
265:        if files:
266:            self.pictures = files
267:
268:        elif not self.pictures and not files:
269:                        #if pictures list contains any file locations
270:                        #and if no files have been specified
271:                        #if it does not, then program will find all the
272:                        #files that are currently in the directory
273:                        #useful if user does not want to overwrite
274:                        #what is currently there
275:
276:            self.pictures.append(self.pictures_dir) #add parent directory
277:                                    #because it will not be
278:                                    #there to start
279:            self.__get_directories_recursively(self.pictures_dir, self.pictures)
280:
281:            self.valid_extensions.append(".ps") #temporarily add postscript to
282:                                    #acceptable extensions so that the
283:                                    #function used will not exclude it
284:                                    #entry is removed later
285:            picture_files = []
286:            for directory in self.pictures: #find all files that are not
287:                                    #subdirectories
288:                for file in self.__find_files(directory):
289:                    picture_files.append(file)
290:
291:            self.valid_extensions.remove(".ps") #remove postscript from
292:                                    #acceptable extensions because
293:                                    #it is no longer needed
294:
295:            self.pictures = picture_files #"self.pictures" now contains
296:                                #locations of files
297:
298:        total_pages = 0
299:        for picture in self.pictures: #iterate through and parse each file
300:                                #looking for a keyword
301:                                #keyword appears twice once with
302:                                #parenthases and once without, so
303:                                #algorithm makes sure it contains no
304:                                #parenthases
305:            file = open(picture, "r")
306:            for line in file.readlines():
307:                if "%%Pages: " in line and "(" not in line:
308:                    total_pages = (total_pages + math.ceil(int(line.split("%%Pages: ")[1].split("\n")[0]) / 2))
309:
310:        return total_pages
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Jun  6 15:20:10 2019

@author: aiden
"""

import os
import treelib

import DirectoryTree


class Print(DirectoryTree.DirectoryTree): #inherits DirectoryTree so that
                                #a directory tree can be made
    """
    contains methods for printing code based on
    "config.txt"
    """
    def __init__(self):
        DirectoryTree.DirectoryTree.__init__(self)

        self.directory_tree = treelib.Tree() #will contain all directories ordered according
                                    #to "config.txt"

        self.swap_tree = treelib.Tree() #used to re-arrange directories in
                                #other node

        self.parent = "." #will contain parent to append to file paths
                        #in "config.txt"
        self.headers_only = 0 #option set in "config.txt"
        self.headers_first = 1 #option set in "config.txt"

        self.dir_order = [] #will contain order of directories set in "config.txt"
        self.file_order = [] #will contain order of files set in "config.txt"

        self.print_order = [] #will contain paths of all print items in order



    @classmethod
    def __get_file_type(cls, path):
        """
        takes a file converted to postscript and finds what file type it
        is (.cpp, .hpp, .sh)

        @params = path name of file to get the type of
        @return = type string of extension (ex. ".cpp")
                type string of root path (ex. no directories and file type (CPP) cut off)
                type string of path without root path
        """
        valid_extensions = [ #files with this extension will be allowed to
                        #be printed
            "PY",
            "C",
            "CPP",
            "HPP",
            "H",
            "SH",
            "TXT",
            "JSON",
            "MD"
            ]

        filename, _ = os.path.splitext(path) #remove extension
        filename = filename.split("/") #remove other directories
        filename = filename[len(filename) - 1]

        one_char = filename[-1:]
        two_char = filename[-2:]
        three_char = filename[-3:]
        four_char = filename[-4:]

        root_name = ""

        if one_char in valid_extensions and two_char not in valid_extensions:
            #added checking for two char as well to fix cases SH and H
            extension = one_char
        elif two_char in valid_extensions:
            extension = two_char
        elif three_char in valid_extensions:
            extension = three_char
        elif four_char in valid_extensions:
            extension = four_char
        else:
            extension = "invalid"

        if extension != "invalid":
            root_name = filename.split(extension)[0]

        root_path = path.split(root_name + extension)[0]

        return extension, root_name, root_path



    def __get_rules(self):
        """
        gets rules based on "config.txt"

        @params = None
        @return = None
        """
        #extract parent from "config.txt"
        with open("config.txt", "r") as config: #file closed at end of block
            for line in config.readlines():
                #makes sure all required elements are in the line
                #and that it is not a comment
                if "parent" in line and "=" in line and "#" not in line:
                    self.parent = line.split("parent")[1].strip().strip('\"')
```

```
114:                self.parent = self.parent.split("=")[1].strip().strip('\"')
115:
116:        #extract other parameters from "config.txt"
117:        param_names = ["HEADERS_FIRST", "ONLY_HEADERS"]
118:        with open("config.txt", "r") as config: #file closed at end of block
119:            for line in config.readlines():
120:                #makes sure all required elements are in the line
121:                #and that it is not a comment
122:                if param_names[0] in line and "#" not in line:
123:                    line = line.split(param_names[0])[1].strip()
124:                    self.headers_first = int(line)
125:
126:                elif param_names[1] in line and "#" not in line:
127:                    line = line.split(param_names[1])[1].strip()
128:                    self.headers_only = int(line.strip())
129:
130:        #read rules
131:        with open("config.txt", "r") as config: #file closed at end of block
132:            for line in config.readlines():
133:                if '#' not in line:
134:                    if "dir" in line:
135:                        #convert to standard os path
136:                        directory = self.parent + "/" + (line.split("dir ")[1].rstrip())
137:                        directory = os.path.normpath(directory)
138:                        self.dir_order.append(directory)
139:                    elif "file" in line:
140:                        #convert to standard os path
141:                        file = self.parent + "/" + (line.split("file ")[1].rstrip())
142:                        file = os.path.normpath(file)
143:                        self.file_order.append(file)
144:
145:
146:
147:
148:    def __order_directories(self):
149:        """
150:        orders directories in "self.directory_tree"
151:        based on settigns in "config.txt"
152:        makes new tree stored in "self.swap_tree"
153:
154:        @params = None
155:        @return = None
156:        """
157:        #iterate through each branch of the tree and sort it
158:        #moving directories to needed spot in "self.swap_tree"
159:        #standard path will be made from os module
160:        #so that no conflicts occur from directory separators
161:        #a two number string is added to all the tags in the node so that
162:        #the order is maintained. The order for nodes is in
163:        #alphabetical order, so no changes would occur if the numbers were
164:        #not added
165:
166:        #finds nodes in tree that have children
167:        parent_nodes = []
168:        for node in self.directory_tree.all_nodes():
169:            if not node.is_leaf(): #a leaf has no children
170:                parent_nodes.append(node)
171:
172:        #adds the root node to the swap tree
173:        self.swap_tree.create_node("00" + parent_nodes[0].tag,
174:                                   parent_nodes[0].identifier)
175:        parent_nodes.pop(0)
176:
177:        #sorts children of parent node
178:        #
179:        #a string of two numbers is added to the tag so that the tree is
180:        #in order, the id is not affected however
181:        #this means the max amount of directories in one node can only be 99
182:        #this should probably not be exceded
183:        for node in parent_nodes:
184:            num = 0
185:            children = self.directory_tree.children(node.identifier)
186:            try: #adds parent of the node to the tree if it is not already there
187:                self.swap_tree.create_node("00" + node.tag,
188:                                           node.identifier,
189:                                           self.directory_tree.parent(node.identifier).identifier)
190:            except treelib.exceptions.DuplicatedNodeIdError: #node already exists
191:                pass
192:
193:            #get applicable rules
194:            applicable_rules = [] #contains identifier for nodes in order to
195:                                  #be sorted
196:            for rule in self.dir_order:
197:                dir_length = len(rule.split("/")) #checks to see if amount of
198:                                                  #directories in path is the
199:                                                  #same as the amount in the
200:                                                  #tree
201:                                                  #if it is, it is added to
202:                                                  #applicable rules
203:
204:                offset = len(self.parent.split("/")) #offset is added to the
205:                                                     #tree level because the
206:                                                     #rules in "config.txt" do
207:                                                     #not contain the parent
208:                                                     #directory
209:                children_names = list(map(lambda x: x.identifier, children))
210:                tree_level = self.directory_tree.level(node.identifier) + offset
211:                if rule in children_names and dir_length == tree_level:
212:                    applicable_rules.append(rule)
213:
214:            for child in applicable_rules: #add nodes to the tree that appear
215:                                           #in the given rules so that the order
216:                                           #wanted is kept
217:                num = str(num) #makes sure that "num" is two characters
218:                if len(num) < 2:
219:                    num = "0" + str(num)
220:
221:                nid = self.directory_tree.get_node(child)
222:                self.swap_tree.create_node(str(num) + nid.tag,
223:                                           nid.identifier,
224:                                           node.identifier)
225:                num = int(num) + 1
226:                children.remove(self.directory_tree.get_node(child))
```

```
227:
228:            while children: #adds other children to the new tree
229:                    #order does not matter so they are added in order
230:                    #of appearance
231:                num = str(num) #makes sure that "num" is two characters
232:                if len(num) < 2:
233:                    num = "0" + num
234:
235:                self.swap_tree.create_node(str(num) + children[0].tag,
236:                                           children[0].identifier,
237:                                           parent=node.identifier)
238:                num = int(num) + 1
239:                children.pop(0)
240:
241:
242:
243:
244:        def __order_headers(self, files):
245:            """
246:            takes a list of postscript files and orders it so that headers
247:            appear before implementation files in the list
248:
249:            @params = list of postscript files to sort putting headers before
250:                    implementation file
251:            @return = sorted list of postscript files
252:            """
253:            #create dictionary with key of the root file name and then a value of
254:            #a list of the extensions with that root name
255:            file_dict = {}
256:
257:            for file in files:
258:                extension, root_name, root_path = self.__get_file_type(file)
259:                if root_name not in file_dict.keys():
260:                    file_dict.update({root_name : [extension]})
261:                else: #if key exists then add value to the list
262:                    file_dict[root_name].append(extension)
263:
264:            #iterate through dictionary placing hpp files ahead of cpp
265:            for extension_list in file_dict.values():
266:                extension_list.sort(reverse=self.headers_first)
267:
268:            #re construct list
269:            ordered_files = []
270:            for item in file_dict:
271:                extensions = file_dict.get(item)
272:                for extension in extensions:
273:                    if extension != "invalid":
274:                        if not self.headers_only:
275:                            file = root_path + item + extension + ".ps"
276:                            ordered_files.append(file)
277:                        else: #if only headers are to be printed
278:                            if extension != "CPP" or len(extensions) == 1:
279:                                file = root_path + item + extension + ".ps"
280:                                ordered_files.append(file)
281:
282:            return ordered_files
283:
284:
285:
286:
287:        def __order_files(self):
288:            """
289:            creates a list of file names in the order that they are to be printed
290:
291:            @params = None
292:            @return = None
293:            """
294:            #get directories to look through
295:            directories = list(self.swap_tree.expand_tree())
296:            ordered_files = [] #holds list of all ordered files
297:
298:            for directory in directories:
299:                ordered_directory = [] #holds list of ordered files in a directory
300:
301:                #gets the files in a given directory
302:                files = list([f.path for f in os.scandir(directory) if not f.is_dir()])
303:                files.sort() #makes paths in aphabetical order
304:
305:                #gets rules for files in a directory
306:                applicable_rules = []
307:
308:                for rule in self.file_order:
309:                    rule_directory = rule
310:                    rule_directory = rule_directory.split("/")
311:                    rule_directory.pop(-1)
312:                    rule_directory = "/".join(rule_directory)
313:
314:                    rule_dirs = len(rule.split("/")) - 1
315:
316:                    #add one to the tree level because the level starts at 0 instead
317:                    #of 1 like the method to find the height of other directories
318:                    tree_level = self.swap_tree.level(self.swap_tree.get_node(directory).identifier) + 1
319:
320:                    #if rule applies
321:                    if rule_directory == directory and rule_dirs == tree_level:
322:                        applicable_rules.append(rule)
323:
324:                #adds files where a specified order has been given
325:                for file in applicable_rules:
326:                    ordered_directory.append(file)
327:                    files.remove(file)
328:
329:                #adds rest of files
330:                while files:
331:                    ordered_directory.append(files[0])
332:                    files.pop(0)
333:
334:                #sorts so that headers are first if that option
335:                #has been given
336:                #also checks if only headers will be printed
337:                ordered_directory = self.__order_headers(ordered_directory)
338:
339:                #adds item in the ordered files in a directory to all the ordered
```

```
340:            #files
341:            for item in ordered_directory:
342:                ordered_files.append(item)
343:
344:
345:        #adds items in ordered_files to the final print order if the
346:        #extension is .ps
347:        for item in ordered_files:
348:            _, extension = os.path.splitext(item)
349:            if extension == ".ps":
350:                self.print_order.append(item)
351:
352:
353:
354:    def order(self):
355:        """
356:        orders code based on "config.txt" with all the parameters set in
357:        it
358:
359:        @params = None
360:        @return = type list of files to print in order
361:        """
362:        #reset attributes
363:        self.directory_tree = treelib.Tree() #will contain all directories ordered according
364:                                             #to "config.txt"
365:        self.swap_tree = treelib.Tree() #used to re-arrange directories in
366:                                        #other node
367:        self.headers_only = 0 #option set in "config.txt"
368:        self.headers_first = 1 #option set in "config.txt"
369:        self.dir_order = [] #will contain order of directories set in "config.txt"
370:        self.file_order = [] #will contain order of files set in "config.txt"
371:        self.print_order = [] #will contain paths of all print items in order
372:
373:        #order code functions
374:        self.__get_rules()
375:        self.directory_tree, _ = self.return_directory_tree(self.parent)
376:
377:        self.__order_directories()
378:        self.__order_files()
379:
380:        return self.print_order
381:
382:
383:
384:
385:    def print_code(self, files=None):
386:        """
387:        prints code by file and also barriers if they are in the list
388:
389:        @params = (optional) type list of path of files to print
390:        @return = None
391:        """
392:
393:        if files:
394:            self.print_order = files
395:
396:        for file in self.print_order: #adds print job to the printer queue
397:                                      #with shell command
398:            print("queuing ", file)
399:            command = "lpr -P HP-Color-LaserJet-M553 -o sides=two-sided-long-edge " + str(file)
400:            os.system(command)
401:
402:    def output_pdf(self, files=None):
403:        """
404:        merges code by file and also barriers if they are in the list
405:
406:        @params = (optional) type list of path of files to print
407:        @return = None
408:        """
409:
410:        if not files:
411:            files - self.print_order
412:
413:        command = "psmerge " + " ".join(files) + " > all.ps; ps2pdf all.ps; rm all.ps"
414:        print(command)
415:        #os.system(command)
416:
```

```
#contains rules for print order
#all files are printed alphabetically unless otherwise specified

#parent will specify which directory to start at. It should be set to where pictures are
parent = ../Pictures


#for rules: algorithm first finds all directories and sorts them according
#to rules. Directory rules starts with "dir" (can be in any order).

#Once all directories and their subdirectories are sorted, algorithm
#will look for individual files rules specifies with "file"

#formatting will look best as a directory tree
#only one rule per line

#order for print goes directory, files in that directory, subdirectory,
#files in the subdirectories of that parent


#set to "1" if user wants headers to be printed before implementation files
HEADERS_FIRST 1

#set to "1" if user only wants header files to be printed to save paper
#this will only keep implementation files that have no header
ONLY_HEADERS 0


#rules

#for rules involving headers and implementation files
#specify both the HPP and CPP file
#by setting HEADERS_FIRST the HPP file will be printed first
#even if the CPP file is listed first

dir Print
dir PIDDebugging
   file PIDDebugging/mainPY.ps
dir Prototypes
dir DocumentationMacros
dir CodeGenerator
   file CodeGenerator/code_genPY.ps
   file CodeGenerator/cpp_typesPY.ps
   file CodeGenerator/exceptionsPY.ps
   file CodeGenerator/configPY.ps
dir AutonSimulator
   file AutonSimulator/mainPY.ps
dir Scouting
   file Scouting/ReadmeMD.ps
dir RobotCode
   file RobotCode/lcdTestSH.ps
   file RobotCode/configJSON.ps
   file RobotCode/stacktraceSH.ps
   dir RobotCode/include
   dir RobotCode/src
      file RobotCode/src/mainCPP.ps
      dir RobotCode/src/objects/controller
      dir RobotCode/src/objects/motors
      dir RobotCode/src/objects/sensors
      dir RobotCode/src/objects/robotChassis
      dir RobotCode/src/objects/lift
      dir RobotCode/src/objects/tilter
      dir RobotCode/src/objects/writer
      dir RobotCode/src/objects/lcdCode
         file RobotCode/src/objects/lcdCode/Debug/DebugHPP.ps
         file RobotCode/src/objects/lcdCode/Debug/DebugCPP.ps
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Jun 11 14:26:00 2019

@author: aiden
"""

import copy
import os

import PostScript
import Print
import DirectoryTree


print("starting print process")
print("please be very careful because there is a lot of code")
print()


help_dict = {
    "help":"print help",
    "postscript":"convert files to postscript",
    "page_count":"prints amount of pages that code will be",
    "order":"prepare order to print in",
    "show_order":"shows current order of files to be printed in",
    "tree":"add cover sheet of tree to files",
    "print":"prints code",
    "pdf":"converts all code into one large pdf",
    "cancel":"cancels all printer jobs",
    "status":"shows current print jobs",
    "exit":"ends session"
    }


ps = PostScript.PostScript()
printer = Print.Print()
Tree = DirectoryTree.DirectoryTree()


print_order = []


while 1:
    command = input("enter command ")

    if command.upper() == "HELP":
        print()

        for item in help_dict:
            string = item
            while len(string) < 20: #aligns help options with each other
                string = string + "."
            print(string, help_dict.get(item), sep=".")


    elif command.upper() == "POSTSCRIPT":
        ps.prepare_code()

    elif command.upper() == "PAGE_COUNT":
        if not print_order:
            pages = ps.get_page_count()
        else:
            pages = ps.get_page_count(print_order)

        print("number of pages: ", pages)

    elif command.upper() == "ORDER":
        if len(print_order) > 1:
            print_order = []

        to_add = printer.order()

        for item in to_add:
            print_order.append(item)
        print()
        print("please review this order carefully before printing")
        print("do not waste paper")
        print()

        for item in print_order:
            print(item)


    elif command.upper() == "SHOW_ORDER":
        for item in print_order:
            print(item)


    elif command.upper() == "TREE":
        file_tree, _ = Tree.return_tree(".")
        filtered_tree = copy.deepcopy(file_tree)

        #sorts out nodes that are not being printed in three ways
        #Tree is copies to a new tree so that the tree does not
        #change during iteration
        #original is looked through and changes are made to the copy
        for node in file_tree.all_nodes():
            nid = node.identifier
            if os.path.isdir(nid):  #sort out nodes by directory
                if nid in ps.directory_exceptions and filtered_tree.get_node(nid):
                    filtered_tree.remove_node(nid)
            else:
                _, extension = os.path.splitext(nid)

                #sort out nodes by file
                if nid in ps.file_exceptions and filtered_tree.get_node(nid):
                    filtered_tree.remove_node(nid)

                #sort out nodes by extension
                elif extension not in ps.valid_extensions and filtered_tree.get_node(nid):
                    filtered_tree.remove_node(nid)
```

```
114:
115:       filtered_tree.get_node("..").tag = "VexCode-2019"
116:       filtered_tree.show()
117:       os.remove("tree.rtf")
118:
119:       filtered_tree.save2file("tree.rtf")
120:       filtered_tree.save2file("tree.txt")
121:       print_order.insert(0, "tree.rtf")
122:
123:
124:    elif command.upper() == "PRINT":
125:       printer.print_code(print_order)
126:
127:    elif command.upper() == "PDF":
128:       printer.output_pdf(print_order)
129:
130:    elif command.upper() == "CANCEL":
131:       os.system("lprm -P HP-Color-LaserJet-M553 -")
132:
133:    elif command.upper() == "STATUS":
134:       os.system("lpstat -P HP-Color-LaserJet-M553")
135:
136:    elif command.upper() == "EXIT":
137:       break
138:
139:    else:
140:       print("invalid command")
141:
142:    print()
143:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jan  5 17:13:31 2020

@author: aiden
"""
import streamlit as st

import data_parser
import graph

# if len(sys.argv) == 1:
#     file = input("enter file to parse: ")
# else:
#     file = sys.argv[1]
file = "./log.txt"

# parser.gen_sample_data()
p = data_parser.Parser()
p.parse_file(file)
p.print_data()
g = graph.DebugGraph(
    p.get_data(),
    {
        "kP":p.get_data()["pid_constants"]["kP"],
        "kI":p.get_data()["pid_constants"]["kI"],
        "kD":p.get_data()["pid_constants"]["kD"],
        "I_max":p.get_data()["pid_constants"]["I_max"],
        "brakemode":p.get_data()["brakemode"],
        "gearset":p.get_data()["gearset"],
        "slew":p.get_data()["slew_rate"]
    }
)
y1 = st.sidebar.selectbox("Y1 data", ["velocity", "voltage", "heading", "position", "integral", "correction", "acceleration"], 1)
track_y1 = st.sidebar.checkbox("Graph Y1 setpoint")
y2 = st.sidebar.selectbox("Y2 data", ["velocity", "voltage", "heading", "position", "None"], 4)
track_y2 = st.sidebar.checkbox("Graph Y2 setpoint")
plot = g.make_graph(y1, y2, track_y1, track_y2)
st.plotly_chart(plot, use_container_width=True)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jan  5 15:35:01 2020

@author: aiden
"""
import math
import random
import os


class Parser:
    """
    parses data from motors so that it can be graphed
    """
    def __init__(self):
        self.__voltage_data = {
            "back_right":[],
            "back_left":[],
            "front_right":[],
            "front_left":[]
        }
        self.__velocity_data = {
            "back_right":[],
            "back_left":[],
            "front_right":[],
            "front_left":[]
        }
        self.__time_data = []
        self.__integral_data = []
        self.__vel_setpoint_data = []
        self.__heading_sp_data = []
        self.__heading_data = []
        self.__position_sp = []
        self.__position_l_data = []
        self.__position_r_data = []

        self.__correction_data = []

        self.__brakemode = None
        self.__gearset = None
        self.__slew = 0
        self.__pid = {
            "kP":0,
            "kI":0,
            "kD":0,
            "I_max":0
            }

        self.__brakemode_names = {
            0:"Coast",
            1:"Brake",
            2:"Hold",

        }

        self.__gearset_names = {
            0:"36:1",
            1:"18:1",
            2:"6:1"
        }


    def __get_data_point(self, line):
        """
        parses a line from the file and returns the data of voltage,
        velocity, integral value, and time in the form of a dictionary

        sample line:
            [INFO] Motor 1, Brakemode: xxxx, Actual_Voltage: xxx, ...
        """
        try:
            line = line.split("[INFO]")[1]
            data = line.split(",")
        except IndexError:  #ensures that line is an actual data line
            return 0
        try:
            voltage1 = [ item.strip().split("Actual_Vol1")[1].split(": ")[1] for item in data if "Actual_Vol1:" in item ]
            voltage2 = [ item.strip().split("Actual_Vol2")[1].split(": ")[1] for item in data if "Actual_Vol2:" in item ]
            voltage3 = [ item.strip().split("Actual_Vol3")[1].split(": ")[1] for item in data if "Actual_Vol3:" in item ]
            voltage4 = [ item.strip().split("Actual_Vol4")[1].split(": ")[1] for item in data if "Actual_Vol4:" in item ]

            velocity1 = [ item.strip().split("Actual_Vel1")[1].split(": ")[1] for item in data if "Actual_Vel1:" in item ]
            velocity2 = [ item.strip().split("Actual_Vel2")[1].split(": ")[1] for item in data if "Actual_Vel2:" in item ]
            velocity3 = [ item.strip().split("Actual_Vel3")[1].split(": ")[1] for item in data if "Actual_Vel3:" in item ]
            velocity4 = [ item.strip().split("Actual_Vel4")[1].split(": ")[1] for item in data if "Actual_Vel4:" in item ]

            time = [ item.strip().split("Time:")[1] for item in data if "Time:" in item ]
            integral = [ item.strip().split("I:")[1] for item in data if "I:" in item ]
            vel_sp = [ item.strip().split("Vel_Sp:")[1] for item in data if "Vel_Sp:" in item ]
            heading_sp = [ item.strip().split("Heading_Sp:")[1] for item in data if "Heading_Sp:" in item ]
            relative_heading = [ item.strip().split("Relative_Heading:")[1] for item in data if "Relative_Heading:" in item ]
            position_sp = [ item.strip().split("Position_Sp:")[1] for item in data if "Position_Sp:" in item ]
            position_r = [ item.strip().split("position_r:")[1] for item in data if "position_r:" in item ]
            position_l = [ item.strip().split("position_l:")[1] for item in data if "position_l:" in item ]

            correction = [ item.strip().split("Correction:")[1] for item in data if "Correction:" in item ]

            data_dict = {
                "voltage1":float(voltage1[0].strip()),
                "voltage2":float(voltage2[0].strip()),
                "voltage3":float(voltage3[0].strip()),
                "voltage4":float(voltage4[0].strip()),
                "velocity1":float(velocity1[0].strip()),
                "velocity2":float(velocity2[0].strip()),
                "velocity3":float(velocity3[0].strip()),
                "velocity4":float(velocity4[0].strip()),
                "time":float(time[0].strip()),
                "integral":float(integral[0].strip()),
                "heading_sp":float(heading_sp[0].strip()),
                "relative_heading":float(relative_heading[0].strip()),
                "position_sp":float(position_sp[0].strip()),
```

```
            "position_r_data":float(position_r[0].strip()),
            "correction":float(correction[0].strip()),
            "position_l_data":float(position_l[0].strip())
            }
    except:
        return 0

    try:
        data_dict.update({"vel_setpoint":float(vel_sp[0].strip())})
    except IndexError:
        data_dict.update({"vel_setpoint":[]})


    return data_dict
    # except IndexError:
    #     return 0


def parse_file(self, file):
    """
    parses a file line by line and adds data to list
    """
    is_first_line = True
    with open(file) as f:
        #find first valid line
        for line in f:
            if is_first_line:
                data = self.__get_data_point(line)
                print(data)
                if data:
                    self.__voltage_data["back_right"].append(data.get("voltage4"))
                    self.__voltage_data["back_left"].append(data.get("voltage3"))
                    self.__voltage_data["front_right"].append(data.get("voltage2"))
                    self.__voltage_data["front_left"].append(data.get("voltage1"))

                    self.__velocity_data["back_right"].append(data.get("velocity4"))
                    self.__velocity_data["back_left"].append(data.get("velocity3"))
                    self.__velocity_data["front_right"].append(data.get("velocity2"))
                    self.__velocity_data["front_left"].append(data.get("velocity1"))

                    self.__time_data.append(data.get("time"))
                    self.__integral_data.append(data.get("integral"))
                    self.__vel_setpoint_data.append(data.get("vel_setpoint"))
                    self.__heading_sp_data.append(data.get("heading_sp"))
                    self.__heading_data.append(data.get("relative_heading"))
                    self.__position_sp.append(data.get("position_sp"))
                    self.__position_r_data.append(data.get("position_r_data"))
                    self.__position_l_data.append(data.get("position_l_data"))

                    self.__correction_data.append(data.get("correction"))

                    first_line = line.split("[INFO]")[1]
                    data = first_line.split(",")

                    self.__brakemode = int([ item.strip().split("Brake:")[1].strip() for item in data if "Brake:" in item ][0])
                    self.__gearset = int([ item.strip().split("Gear:")[1].strip() for item in data if "Gear:" in item ][0])
                    self.__slew = int([ int(item.strip().split("Slew:")[1].strip()) for item in data if "Slew:" in item ][0])
                    # self.__slew = 120


                    self.__brakemode = self.__brakemode_names.get(self.__brakemode, "???")
                    self.__gearset = self.__gearset_names.get(self.__gearset, "???")

                    self.__pid["kP"] = float([ float(item.strip().split("kP:")[1].strip()) for item in data if "kP:" in item ][0])
                    self.__pid["kI"] = float([ float(item.strip().split("kI:")[1].strip()) for item in data if "kI:" in item ][0])
                    self.__pid["kD"] = float([ float(item.strip().split("kD:")[1].strip()) for item in data if "kD:" in item ][0])
                    self.__pid["I_max"] = float([ float(item.strip().split("I_max:")[1].strip()) for item in data if "I_max:" in item ][0])
                    is_first_line = False
                continue


            data = self.__get_data_point(line)
            if data:
                self.__voltage_data["back_right"].append(data.get("voltage4"))
                self.__voltage_data["back_left"].append(data.get("voltage3"))
                self.__voltage_data["front_right"].append(data.get("voltage2"))
                self.__voltage_data["front_left"].append(data.get("voltage1"))

                self.__velocity_data["back_right"].append(data.get("velocity4"))
                self.__velocity_data["back_left"].append(data.get("velocity3"))
                self.__velocity_data["front_right"].append(data.get("velocity2"))
                self.__velocity_data["front_left"].append(data.get("velocity1"))

                self.__time_data.append(data.get("time"))
                self.__integral_data.append(data.get("integral"))
                self.__vel_setpoint_data.append(data.get("vel_setpoint"))
                self.__heading_sp_data.append(data.get("heading_sp"))
                self.__heading_data.append(data.get("relative_heading"))
                self.__position_sp.append(data.get("position_sp"))
                self.__position_r_data.append(data.get("position_r_data"))
                self.__position_l_data.append(data.get("position_l_data"))
                self.__correction_data.append(data.get("correction"))

def print_data(self):
    """
    prints data

    useful for debugging
    """
    print("\nVoltage Data:", len(self.__voltage_data), "data points")
    for item in self.__voltage_data:
        print(item)

    print("\nVelocity Data:", len(self.__velocity_data), "data points")
    for item in self.__velocity_data:
        print(item)

    print("\nIntegral Data:", len(self.__integral_data), "data points")
    for item in self.__integral_data:
        print(item)
```

```
227:         print("\nTime Data:", len(self.__time_data), "data points")
228:         for item in self.__time_data:
229:             print(item)
230:
231:         print("\nVelocity Setpoint Data:", len(self.__vel_setpoint_data), "data points")
232:         for item in self.__vel_setpoint_data:
233:             print(item)
234:
235:         print("\nHeading Setpoint Data:", len(self.__heading_sp_data), "data points")
236:         for item in self.__heading_sp_data:
237:             print(item)
238:
239:         print("\Relative Heading Data:", len(self.__heading_data), "data points")
240:         for item in self.__heading_data:
241:             print(item)
242:
243:         print("\Position Setpoint:", len(self.__position_sp), "data points")
244:         for item in self.__position_sp:
245:             print(item)
246:
247:         # print("\Position Data:", len(self.__position_data), "data points")
248:         # for item in self.__position_data:
249:         #     print(item)
250:
251:
252:         print("\nPID constants:")
253:         for key in self.__pid:
254:             print(key, ":", self.__pid[key])
255:
256:
257:         print("\nBrakemode: ", self.__brakemode)
258:         print("Gearset: ", self.__gearset)
259:         print("Slew Rate: ", self.__slew)
260:
261:
262:     def get_data(self):
263:         """
264:         returns a dictionary of the data that was parsed
265:         """
266:         data = {
267:             "voltage":self.__voltage_data,
268:             "velocity":self.__velocity_data,
269:             "integral":self.__integral_data,
270:             "vel_setpoint":self.__vel_setpoint_data,
271:             "heading_setpoint":self.__heading_sp_data,
272:             "heading_data":self.__heading_data,
273:             "position_sp":self.__position_sp,
274:             "position_r":self.__position_r_data,
275:             "position_l":self.__position_l_data,
276:             "time":self.__time_data,
277:             "brakemode":self.__brakemode,
278:             "gearset":self.__gearset,
279:             "slew_rate":self.__slew,
280:             "correction":self.__correction_data,
281:             "pid_constants":self.__pid
282:         }
283:
284:         return data
285:
286:
287:
288: def gen_sample_data(num_data_pts=1000, setpoint=100, file="ut.txt"):
289:     """
290:     makes a random set of data that can be used for a unit test
291:     """
292:     if os.path.isfile(file):
293:         os.remove(file)
294:     f = open(file, "a")
295:
296:     step = setpoint / num_data_pts #setpoint / ...
297:     min_vel = 0
298:     for i in range(num_data_pts):
299:         vel = (random.randint(int(min_vel), int(min_vel + step)) ** 1/((i+setpoint)/num_data_pts)) + setpoint/10
300:         vol = ((((vel + 200) * (12000 + 12000)) / (200 + 200)) - 12000) + random.randint(-600, 600); #scale vel to voltage range and add jitter
301:         data = "[INFO] Motor 1, Actual_Vol: " + str(vol)
302:         data += ", Brake: 1, Gear: 1, I_max: 1000, I: 100, kD: 0, kI: 0, "
303:         data += "kP: 1.0, Slew: 40, Time: " + str(i)
304:         data += ", Heading_Sp: 0.0, Relative_Heading: " + str((0 + (random.randrange(-100, 100, 1) / 100)))
305:         data += ", Position_Sp: 1000, Position: 1000"
306:         data += ", Vel_Sp: " + str(setpoint) + ", Vel: " + str(vel) + "\n"
307:
308:         f.write(data)
309:
310:         min_vel += step
311:
312:
313:     f.close()
314:
315: # unit test
316:
317: # gen_sample_data(file="test.txt")
318: # P = Parser()
319: # P.parse_file("test.txt")
320: # P.print_data()
321:
322:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 29 12:38:16 2019

@author: aiden
"""
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import numpy as np
import math

class DebugGraph:
    """
    class for making a graph for debugging PID data
    """
    def __init__(self, data_dict, parameters):
        self.time_data = data_dict.get("time")
        self.vel_data = data_dict.get("velocity")
        self.vol_data = data_dict.get("voltage")
        self.vel_sp = data_dict.get("vel_sp")
        self.heading_sp = data_dict.get("heading_sp")
        self.heading_data = data_dict.get("heading_data")
        self.position_sp = data_dict.get("position_sp")
        self.position_r_data = data_dict.get("position_r")
        self.position_l_data = data_dict.get("position_l")
        self.integral_data = data_dict.get("integral")
        self.correction_data = data_dict.get("correction")

        #add legend for constants
        self.constants_text = "kP : " + str(parameters.get("kP")) + "\n"
        self.constants_text += "kI : " + str(parameters.get("kI")) + "\n"
        self.constants_text += "kD: " + str(parameters.get("kD")) + "\n"
        self.constants_text += "Integral Max: " + str(parameters.get("I_max")) + "\n"
        self.constants_text += "\n"
        self.constants_text += "Brakemode: " + str(parameters.get("brakemode")) + "\n"
        self.constants_text += "Gearset: " + str(parameters.get("gearset")) + "\n"
        self.constants_text += "Slew Rate (mV/ms): " + str(parameters.get("slew")) + "\n"

    def __to_in_per_sec2(self, degrees_per_ms2, wheel_diameter=3.25):
        circumference = math.pi * wheel_diameter
        revolutions_per_ms2 = degrees_per_ms2 / 360
        linear_distance = revolutions_per_ms2 * circumference
        in_per_sec2 = linear_distance * 1000  # convert ms to sec
        return in_per_sec2

    def make_graph(self, y1, y2=None, track_y1_sp=True, track_y2_sp=True):
        """
        returns a graph object with given axis parameters
        """
        title = "PID Tuning - "
        x = self.time_data
        if y1 == "velocity":
            y1_data = [self.vel_data.get("back_right"), self.vel_data.get("front_right"), self.vel_data.get("back_left"), self.vel_data.get("front_left")]
            y1_sp = self.vel_sp
            y1_title = "Velocity (RPM)"
            name1 = ["Back Right Velocity", "Front Right Velocity", "Back Left Velocity", "Front Left Velocity"]
            title += "Velocity"
        elif y1 == "voltage":
            y1_data = [self.vol_data.get("back_right"), self.vol_data.get("front_right"), self.vol_data.get("back_left"), self.vol_data.get("front_left")]
            y1_sp = []
            y1_title = "Voltage (mV)"
            name1 = ["Back Right Voltage", "Front Right Voltage", "Back Left Voltage", "Front Left Voltage"]
            title += "Voltage"
        elif y1 == "heading":
            y1_data = [self.heading_data]
            y1_sp = self.heading_sp
            y1_title = "Relative Heading (Degrees)"
            name1 = ["Relative Heading of Robot"]
            title += "Relative Heading of Robot"
        elif y1 == "position":
            y1_data = [self.position_l_data, self.position_r_data]
            y1_sp = self.position_sp
            y1_title = "Position"
            name1 = ["Position of Right Sensor", "Position of Left Sensor"]
            title += "Position of Sensor"
        elif y1 == "integral":
            y1_data = [self.integral_data]
            y1_title = "Integral"
            y1_sp = []
            name1 = ["Integral Value"]
            title += "Integral"
        elif y1 == "correction":
            y1_data = [self.correction_data]
            y1_title = "correction"
            y1_sp = []
            name1 = ["Correction"]
            title += "Correction"
        elif y1 == "acceleration":
            vel_data_l = np.diff(self.position_l_data) / np.diff(x)
            vel_data_r = np.diff(self.position_r_data) / np.diff(x)
            accel_data_l = []
            accel_data_r = []
            for l, r in zip(np.diff(vel_data_l) / np.diff(x[:-1]), np.diff(vel_data_r) / np.diff(x[:-1])):
                accel_data_l.append(self.__to_in_per_sec2(l))
                accel_data_r.append(self.__to_in_per_sec2(r))
            y1_data = [accel_data_l, accel_data_r]
            x = x[:-2]
            y1_title = "Acceleration"
            y1_sp = []
            name1 = ["Acceleration of Right Sensor", "Acceleration of Left Sensor"]
            title += "Acceleration"
        else:
            y1_data = []
            y1_sp = []
            y1_title = ""
            name1 = ""


        if y2 == "velocity":
            y2_data = [self.vel_data.get("back_right"), self.vel_data.get("front_right"), self.vel_data.get("back_left"), self.vel_data.get("front_left")]
            y2_sp = self.vel_sp
            y2_title = "Velocity (RPM)"
```

```
114:            name2 = ["Back Right Velocity", "Front Right Velocity", "Back Left Velocity", "Front Left Velocity"]
115:            title += "Velocity"
116:        elif y2 == "voltage":
117:            y2_data = [self.vol_data.get("back_right"), self.vol_data.get("front_right"), self.vol_data.get("back_left"), self.vol_data.get("front_left")]
118:            y2_sp = []
119:            y2_title = "Voltage (mV)"
120:            name2 = ["Back Right Voltage", "Front Right Voltage", "Back Left Voltage", "Front Left Voltage"]
121:            title += "Voltage"
122:        elif y2 == "heading":
123:            y2_data = [self.heading_data]
124:            y2_sp = self.heading_sp
125:            y2_title = "Relative Heading (Degrees)"
126:            name2 = ["Relative Heading of Robot"]
127:            title += "Relative Heading of Robot"
128:        elif y2 == "position":
129:            y2_data = [self.position_l_data, self.position_r_data]
130:            y2_sp = self.position_sp
131:            y2_title = "Position"
132:            name2 = ["Position of Right Sensor", "Position of Left Sensor"]
133:            title += "Position of Sensor"
134:        elif y2 == "integral":
135:            y2_data = [self.integral_data]
136:            y2_sp = []
137:            name2 = ["Integral Value"]
138:            title += "Integral"
139:        else:
140:            y2_data = []
141:            y2_sp = []
142:            y2_title = ""
143:            name2 = ""
144:
145:        if y2_data:
146:            plot = make_subplots(specs=[[{"secondary_y": True}]])
147:        else:
148:            plot = go.Figure()
149:
150:        for data, name in zip(y1_data, name1):
151:            plot.add_trace(
152:                go.Scatter(
153:                    x=x,
154:                    y=data,
155:                    mode="lines",
156:                    line={'dash': 'solid', 'color': 'blue'},
157:                    name=name,
158:                    yaxis="y1",
159:                )
160:            )
161:        if track_y1_sp:
162:            plot.add_trace(
163:                go.Scatter(
164:                    x=x,
165:                    y=y1_sp,
166:                    mode="lines",
167:                    line={'dash': 'dash', 'color': 'blue'},
168:                    name="Setpoint",
169:                    yaxis="y1",
170:                )
171:            )
172:
173:
174:        if y2_data:
175:            for data, name in zip(y2_data, name2):
176:                plot.add_trace(
177:                    go.Scatter(
178:                        x=x,
179:                        y=data,
180:                        mode="lines",
181:                        line={'dash': 'solid', 'color': 'green'},
182:                        name=name,
183:                    ),
184:                    secondary_y=True
185:                )
186:            if track_y2_sp:
187:                plot.add_trace(
188:                    go.Scatter(
189:                        x=x,
190:                        y=y2_sp,
191:                        mode="lines",
192:                        line={'dash': 'dash', 'color': 'green'},
193:                        name="Setpoint",
194:                    ),
195:                    secondary_y=True
196:                )
197:
198:        title += " vs Time"
199:        plot.update_layout(
200:            title=title,
201:            xaxis=dict(
202:                title="Time (ms)"
203:            ),
204:            font=dict(
205:                family="Courier New, monospace",
206:                size=14,
207:                color="#7f7f7f"
208:            ),
209:            showlegend=True,
210:            legend=dict(x=1.2, y=1.1),
211:            plot_bgcolor="rgb(255, 255, 255)"
212:        )
213:
214:
215:        plot.update_xaxes(
216:            tickangle=-45,
217:            showgrid=False,
218:            mirror=True,
219:            ticks='outside',
220:            showline=True
221:
222:        )
223:        plot.update_yaxes(
224:            tickangle=0,
225:            showgrid=False,
226:            zeroline=True,
```

```
227:             zerolinewidth=2,
228:             zerolinecolor="black",
229:             mirror=True,
230:             ticks='outside',
231:             showline=True
232:
233:         )
234:
235:         if y2_data:
236:             plot.update_yaxes(title_text=y1_title, secondary_y=False, color="blue")
237:             plot.update_yaxes(title_text=y2_title, secondary_y=True, color="green")
238:         else:
239:             plot.update_yaxes(title_text=y1_title)
240:             pass
241:
242:         return plot
243:
244:
245: #unit test
246: #
247: #import numpy as np
248: #
249: #time = range(50)
250: #velocity = [20 * np.sin(x) for x in time]
251: #voltage = [x * 60 for x in velocity]
252: #pid = {"kP":1.0,"kI":.001,"kD":.25,"kI_max":5,"brakemode":"Brake","gearset":"18:1","slew":400}
253: #x = DebugGraph(time, velocity, voltage, 12, pid)
254: #x.get_graph().show()
255: #x.get_graph().savefig("test.png", bbox_inches='tight')
256:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Jan  6 22:38:31 2020

@author: aiden
"""
class LCD:
    """
    class for communicating with the vex lcd via serial communication
    """
    def __init__(self, serial_write, serial_read):
        self.lcd_screen = serial_write
        self.lcd_buttons = serial_read

        self.__flags = 0x00
        self.__line_1 = " " * 16
        self.__line_2 = " " * 16

        self.__num_chars = 16


    def __write_line(self):
        """
        Writes to the lcd. Runs from thread, not called by user

        Returns
        -------
        int
            1 on success.

        """
        #line 1 bytearray
        send_array = bytearray()
        send_array.append(0xAA)
        send_array.append(0x55)
        send_array.append(0x1E)
        send_array.append(0x12)
        send_array.append(0x00)
        checksum = 0x00

        for char in self.__line_1:
            send_array.append(ord(char))
            checksum += (ord(char))

        self.lcd_screen.write(send_array)

        #line 2 bytearray
        send_array = bytearray()
        send_array.append(0xAA)
        send_array.append(0x55)
        send_array.append(0x1E)
        send_array.append(0x12)
        send_array.append(0x01)
        checksum = 0x01

        for char in self.__line_1:
            send_array.append(ord(char))
            checksum += (ord(char))

        self.lcd_screen.write(send_array)

        return 1



    def write_string(self, string, ln=0, align="left"):
        """
        writes a string to the lcd
        handles multiline writing by keeping track of line one and line two
        and adding a newline character between them

        returns 0 on unsucessful write and 1 if completed successfully
        """
        buffer = self.__num_chars - len(string)
        if align == "center":
            left_spaces = round(buffer/2, 0)
            right_spaces = self.__num_chars - len(string) - left_spaces
            if right_spaces < 0:
                right_spaces = 0
            string = (" " * left_spaces) + string + (" " * right_spaces)

        elif align == "right":
            left_spaces = self.__num_chars - len(string)
            if left_spaces < 0:
                left_spaces = 0
            string = (" " * left_spaces) + string

        elif align == "left":
            string = string

        else:
            return 0

        if len(string) > self.__num_chars:  #cap string length to the number of characters on the lcd
            string = string[:self.__num_chars]


        if ln == 0:
            self.__line_1 = string
        elif ln == 1:
            self.__line_2 = string

        return 1



    def clear(self, line):
        """
        clears a line on the lcd by sending spaces

```

```
114:        Parameters
115:        ----------
116:        line : int
117:            the line to clear.
118:
119:        Returns
120:        -------
121:        int
122:            1 on success, 0 on failure.
123:
124:        """
125:        string = " " * 16
126:        ret = self.write_string(string, ln=line)
127:
128:        if not ret:
129:            return 0
130:
131:        return 1
132:
133:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 13 21:14:17 2019

@author: aiden
"""

import time
import pyautogui

time.sleep(.5)

pyautogui.typewrite("/**")
pyautogui.press("enter")

pyautogui.typewrite(" * @param:")
pyautogui.press("enter")

pyautogui.typewrite("* @param:")
pyautogui.press("enter")

pyautogui.typewrite("* @return:")
pyautogui.press("enter")

pyautogui.typewrite("**")
pyautogui.press("enter")

pyautogui.typewrite("* @see:")
pyautogui.press("enter")

pyautogui.typewrite("* @see:")
pyautogui.press("enter")

pyautogui.typewrite("**")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_function_line_1")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_function_line_2")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_function_line_3")
pyautogui.press("enter")

pyautogui.typewrite("**")
pyautogui.press("enter")

pyautogui.typewrite("*/")
pyautogui.press("enter")

pyautogui.press("backspace")
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 13 21:20:16 2019

@author: aiden
"""

import time
import pyautogui

time.sleep(.5)

pyautogui.typewrite("/**")
pyautogui.press("enter")

pyautogui.typewrite(" * how_function_works_line_1")
pyautogui.press("enter")

pyautogui.typewrite("* how_function_works_line_2")
pyautogui.press("enter")

pyautogui.typewrite("* how_function_works_line_3")
pyautogui.press("enter")


pyautogui.typewrite("*/")
pyautogui.press("enter")

pyautogui.press("backspace")
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 13 21:23:07 2019

@author: aiden
"""

import time
import pyautogui

time.sleep(.5)

pyautogui.typewrite("/**")
pyautogui.press("enter")

pyautogui.typewrite(" * @see:")
pyautogui.press("enter")

pyautogui.typewrite("* @see:")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("* purpose_of_class_line_1")
pyautogui.press("enter")

pyautogui.typewrite("* purpose_of_class_line_2")
pyautogui.press("enter")

pyautogui.typewrite("* purpose_of_class_line_3")
pyautogui.press("enter")

pyautogui.typewrite("*/")
pyautogui.press("enter")

pyautogui.press("backspace")
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 13 20:50:03 2019

@author: aiden
"""

import time
import pyautogui

time.sleep(.5)

pyautogui.typewrite("/**")
pyautogui.press("enter")

pyautogui.typewrite(" * @file:")
pyautogui.press("enter")

pyautogui.typewrite("* @author:")
pyautogui.press("enter")

pyautogui.typewrite("* @reviewed_on:")
pyautogui.press("enter")

pyautogui.typewrite("* @reviewed_by:")
pyautogui.press("enter")

pyautogui.typewrite("* TODO:")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_contents_line_1")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_contents_line_2")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_contents_line_3")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("*/")
pyautogui.press("enter")

pyautogui.press("backspace")
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Aug 15 10:16:03 2019

@author: aiden
"""
import inspect
import colorama
import fcntl
import os
import readline
import struct
import termios

import cpp_types
import config
import exceptions


class HeaderGen:
    """
    Wrapper class for working with cpp types and header files
    used to easily generate code based on user input
    """
    def __init__(self, header_obj):
        self.header = header_obj
        self.children = header_obj.get_children()
        self.focus = header_obj
        self.current_type = "header"
        self.loc = "./" + self.header.file_name

        self.commands = {
            "ls":self.__ls,
            "view":self.__view,
            "cd":self.__change_focus,
            "exit":self.__exit,
            "new":self.__new,
            "write":self.__write,
            "add":self.__add,
            "help":self.__help
            }


    @classmethod
    def __exit(cls, *_):
        """
        exit function for shell
        """
        raise exceptions.Exit

    def __ls(self, *_):
        """
        lists data on focused object
        """
        print(self.focus.list_data() + "\n")

    def __view(self, *args):
        """
        shows the generated text of the focused object
        param bool header_text sets the view to either
        the generated header text or the generated implementation
        file text
        """
        if args[0] == []:
            header_text = True
        elif str(args[0][0]).upper() == "HEADER":
            header_text = True
        elif str(args[0][0]).upper() == "IMPLEMENTATION":
            header_text = False
        else:
            raise exceptions.UnknownOption

        text = self.focus.gen_header_text() if header_text else self.focus.gen_impl_text()
        print(text + "\n")

    def __change_focus(self, *args):
        """
        changes focus to user specified input
        no return
        """
        name = args[0][0]
        if name == "..":
            self.focus = self.focus.parent
            self.update_type(self.focus)
            self.children = self.focus.get_children()
        elif self.focus.has_children:
            if name in self.children.keys():
                self.focus = self.children.get(name)
                self.update_type(self.focus)
                self.children = self.focus.get_children()
            else:
                print("invalid selection")
        else:
            print("focused object has no children")


        path = [self.focus]
        while len(path) == len(set(path)):
            path.insert(0, path[0].parent)
        path.pop(0)
        path.pop(0)
        self.loc = "./" + self.header.file_name + "/" + "/".join(x.name for x in path)



    def __new_class(self, name):
        """
        adds a class object to a header file object
        param name = type str of name of the class

        throws invalid addition if not currently focused on header object
        """
```

```
114:         if self.current_type == "header":
115:             obj = cpp_types.cpp_class(name, self.focus)
116:             self.header.classes.append(obj)
117:         else:
118:             raise exceptions.InvalidAddition
119:
120:
121:     def __new_func(self, loc, return_type, name, static=False):
122:         """
123:         adds a function object to a header file or class object
124:         param name = type str of name of the function
125:         param return_type = type of function
126:         param static = is the function static or not
127:
128:         throws invalid addition if incorrect params are passed
129:         """
130:         obj = cpp_types.cpp_func(name, return_type, static, self.focus)
131:
132:         if self.current_type == "header":
133:             self.focus.funcs.append(obj)
134:         elif self.current_type == "class":
135:             if static:
136:                 if loc.upper() == "PUBLIC":
137:                     self.focus.public["static_func"].append(obj)
138:                 elif loc.upper() in ["PROT", "PROTECTED"]:
139:                     self.focus.protected["static_func"].append(obj)
140:                 else:
141:                     self.focus.private["static_func"].append(obj)
142:             else:
143:                 if loc.upper() == "PUBLIC":
144:                     self.focus.public["func"].append(obj)
145:                 elif loc.upper() in ["PROT", "PROTECTED"]:
146:                     self.focus.protected["func"].append(obj)
147:                 else:
148:                     self.focus.private["func"].append(obj)
149:         else:
150:             raise exceptions.InvalidAddition
151:
152:     def __new_var(self, loc, var_type, name, static=False):
153:         """
154:         adds a function object to a header file or class object
155:         param name = type str of name of the function
156:         param var_type = type of variable
157:         param static = is the variable static or not
158:
159:         throws invalid addition if incorrect params are passed
160:         """
161:         obj = cpp_types.cpp_variable(name, var_type, static, self.focus)
162:
163:         if self.current_type == "header":
164:             self.focus.static_vars.append(obj)
165:         elif self.current_type == "class":
166:             if static == "static":
167:                 if loc.upper() == "PUBLIC":
168:                     self.focus.public["static_var"].append(obj)
169:                 elif loc.upper() in ["PROT", "PROTECTED"]:
170:                     self.focus.protected["static_var"].append(obj)
171:                 else:
172:                     self.focus.private["static_var"].append(obj)
173:             else:
174:                 if loc.upper() == "PUBLIC":
175:                     self.focus.public["var"].append(obj)
176:                 elif loc.upper() in ["PROT", "PROTECTED"]:
177:                     self.focus.protected["var"].append(obj)
178:                 else:
179:                     self.focus.private["var"].append(obj)
180:         else:
181:             raise exceptions.InvalidAddition
182:
183:
184:
185:     def __new_include(self, include_type, name):
186:         """
187:         adds an include to a header file
188:         param type (lib, user) - type of include
189:         param name - name of include
190:
191:         throws invalid addition if incorrect params are passed
192:         """
193:         if include_type == "user":
194:             self.header.user_includes.append(name.strip())
195:         elif include_type == "lib":
196:             self.header.lib_includes.append(name.strip())
197:         else:
198:             raise exceptions.InvalidAddition
199:
200:
201:
202:     def __new(self, *args):
203:         """
204:         adds a type of object to another type of object
205:         checks to see if addition is valid
206:         ex. if creating a class the parent must be of type header
207:
208:         throws InvalidAddition if the addition failed
209:         """
210:         func_dict = {
211:             "class":self.__new_class,
212:             "var":self.__new_var,
213:             "func":self.__new_func,
214:             "function":self.__new_func,
215:             "include":self.__new_include
216:             }
217:
218:         if not args[0]:
219:             raise exceptions.InvalidAddition("invalid parameters were passed")
220:
221:         obj_type = args[0][0].strip()
222:         params = list(map(lambda x: x.strip(), args[0][1:]))
223:
224:         func = func_dict.get(obj_type)
225:         if func:
226:             num_args = len(inspect.signature(func).parameters)
```

```
227:            if num_args > len(params):
228:                params.insert(0, "")
229:            while num_args > len(params):
230:                params.append("")
231:
232:        elif func_dict.get(args[0][1].strip()): #if argument in second position
233:                                    #is a valid command then switch
234:                                    #param in first position to front
235:                                    #of params list
236:            func = func_dict.get(args[0][1].strip())
237:            params = list(map(lambda x: x.strip(), args[0][2:]))
238:
239:            num_args = len(inspect.signature(func).parameters)
240:            if num_args > len(params):
241:                params.insert(0, args[0][0].strip())
242:            while num_args > len(params):
243:                params.append("")
244:
245:        else:
246:            raise exceptions.InvalidAddition("invalid function call")
247:
248:        func(*params[:num_args])
249:        self.children = self.focus.get_children()
250:
251:
252:
253:    def __add_function_param(self, param_type, param_name):
254:        """
255:        adds parameters to a function
256:        @param param_type - type str of the cpp type
257:        @param param_name - name of the parameter
258:        throws Invalid Addition on failure or invalid params
259:        """
260:        if self.current_type != "function":
261:            raise exceptions.InvalidAddition
262:
263:        param = param_type.strip() + " " + param_name.strip()
264:        self.focus.params.append(param)
265:
266:
267:    def __add(self, *args):
268:        """
269:        used to add attributes such as parameters to a function
270:
271:        throws invalid addition if the addition failed
272:        """
273:        func_dict = {
274:            "param":self.__add_function_param
275:            }
276:
277:        if not args[0]:
278:            raise exceptions.InvalidAddition("invalid parameters were passed")
279:
280:        obj_type = args[0][0].strip()
281:        params = list(map(lambda x: x.strip(), args[0][1:]))
282:
283:        func = func_dict.get(obj_type)
284:
285:        if func:
286:            num_args = len(inspect.signature(func).parameters)
287:            while num_args > len(params):
288:                params.append("")
289:
290:            func(*params[:num_args])
291:            self.children = self.focus.get_children()
292:
293:        else:
294:            raise exceptions.InvalidAddition("invalid function call")
295:
296:
297:
298:
299:    def __write(self, *_):
300:        """
301:        writes the text generated from the header file into an actual file
302:        as well as generating and writing the text for an implementation
303:        file
304:        """
305:        header_file_name = self.header.file_name
306:        impl_file_name, _ = os.path.splitext(self.header.file_name)
307:        impl_file_name += ".cpp"
308:
309:        with open(header_file_name, "a") as file:
310:            file.write(self.header.gen_header_text())
311:
312:        with open(impl_file_name, "a") as file:
313:            file.write(self.header.gen_impl_text())
314:
315:
316:
317:
318:    def __help(self, *_):
319:        """
320:        prints docstrings for each function
321:        """
322:        #get terminal size to set max chars per line
323:        _, columns, _, _ = struct.unpack('HHHH',
324:                                         fcntl.ioctl(0, termios.TIOCGWINSZ,
325:                                         struct.pack('HHHH', 0, 0, 0, 0)))
326:
327:        max_chars = columns - 5
328:        help_msg = ""
329:        spaces = len(max(list(self.commands.keys()), key=len))
330:
331:        for key in self.commands:
332:            doc_str = str(self.commands.get(key).__doc__).strip().replace("\n", " ")
333:            words = doc_str.split(" ")
334:            words = list(filter(lambda a: a != "", words))
335:
336:            line = key + (" " * (spaces - len(key))) + "  -  "
337:            indentation = len(line) * " "
338:            for word in words:
339:                if (len(word) + len(line)) < max_chars:
```

```
340:                 line += word + " "
341:             else:
342:                 help_msg += line + "\n"
343:                 line = indentation + word + " "
344:         help_msg += line + "\n\n"
345:
346:     print(help_msg)
347:
348:
349:
350:
351: def update_type(self, obj):
352:     """
353:     updates self.current_type to the type of obj
354:     """
355:     types = {cpp_types.cpp_class:"class",
356:             cpp_types.cpp_func:"function",
357:             cpp_types.HeaderFile:"header",
358:             cpp_types.cpp_variable:"variable"
359:             }
360:     self.current_type = types.get(type(obj))
361:
362:
363:
364:
365: def execute_command(self, command):
366:     """
367:     executes a command from the given api
368:     api commands are stored in self.commands
369:     """
370:     cmd = self.commands.get(command.split(" ")[0].strip())
371:     args = command.split(" ")[1:]
372:     for arg in args:
373:         arg.strip()
374:
375:     try:
376:         if not cmd:
377:             raise exceptions.UnknownOption
378:         cmd(args)
379:     except exceptions.UnknownOption:
380:         pass
381:     except exceptions.InvalidAddition:
382:         pass
383:
384:
385:
386: #TODO: add dynamically changing autocomplete
387: class Shell:
388:     """
389:     contains shell like interface for generating code
390:     """
391:     def __init__(self):
392:         self.loc = "./"
393:         self.functions = sorted(["new",
394:                                  "class",
395:                                  "edit",
396:                                  "ls",
397:                                  "include",
398:                                  "user",
399:                                  "lib",
400:                                  "func",
401:                                  "view",
402:                                  "exit",
403:                                  "add"])
404:
405:
406:     def auto_complete(self, text, state):
407:         """
408:         function that will attempt to autocomplete user input
409:         on <tab> to s member in self.functions
410:         returns type str of first matched string
411:         """
412:         if state == 0:  # on first trigger, build possible matches
413:             if text:  # cache matches (entries that start with entered text)
414:                 matches = [s for s in self.functions if s and s.startswith(text)]
415:             else:  # no text entered, all matches possible
416:                 matches = self.functions[:]
417:
418:         # return match indexed by state
419:         try:
420:             return matches[state]
421:         except IndexError:
422:             return None
423:
424:
425:     def get_command(self):
426:         """
427:         gets command from user
428:         returns type str of command
429:         """
430:         print("")
431:
432:         colorama.init()
433:         readline.set_completer(self.auto_complete)
434:         readline.parse_and_bind('tab: complete')
435:
436:         command = input((config.SHELL_COLOR
437:                          + "\n"
438:                          + config.CURSOR_UP_ONE
439:                          + config.ERASE_LINE
440:                          + self.loc + " <command> "
441:                          + colorama.Fore.RESET))
442:         return command
443:
444:
445:
446: #def unit_test():
447: #    h = cpp_types.HeaderFile("MyHeader.hpp")
448: #
449: #    c = cpp_types.cpp_class("MyClass")
450: #    c.protected["func"].append(cpp_types.cpp_func("my_func", "int", 1))
451: #    c.public["var"].append("int x")
452: #    h.static_vars.append("int y")
```

```
453:  #    h.classes.append(c)
454:  #
455:  #    t = h.gen_header_text()
456:  #    print(t)
457:
458:
459:
460:
461:  file_name = input(config.SHELL_COLOR
462:                    + "\n"
463:                    + config.CURSOR_UP_ONE
464:                    + config.ERASE_LINE
465:                    + "Enter name of Header File to Create "
466:                    + colorama.Fore.RESET)
467:
468:  if not any(s in file_name and s[-len(s):] == file_name[-len(s):] for s in ['.hpp', '.h']):
469:      raise exceptions.InvalidFileName
470:
471:
472:
473:  header = cpp_types.HeaderFile(file_name)
474:  s = Shell()
475:  s.loc += file_name + "/"
476:  header_gen = HeaderGen(header)
477:
478:
479:  while 1:
480:      try:
481:          try:
482:              usr_command = s.get_command()
483:          except KeyboardInterrupt:
484:              print()
485:              raise exceptions.Exit
486:
487:          header_gen.execute_command(usr_command)
488:          s.loc = header_gen.loc
489:
490:      except exceptions.Exit:
491:          break
492:
493:
494:
495:  #unit_test()
496:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Aug 17 19:52:41 2019

@author: aiden
"""
import os
import config

class cpp_class:
    """
    contains methods and data for a cpp class
    generates text for header and implementation files
    """
    def __init__(self, name, parent):
        self.name = name

        self.has_children = True
        self.parent = parent

        self.private = {
            "func":[],
            "static_func":[],
            "var":[],
            "static_var":[]
            }
        self.protected = {
            "func":[],
            "static_func":[],
            "var":[],
            "static_var":[]
            }
        self.public = {
            "func":[],
            "static_func":[],
            "var":[],
            "static_var":[]
            }



    @classmethod
    def __keys_empty(cls, section):
        """
        checks if all keys in a dictionary have no value
        returns True if empty, False otherwise
        """
        for key in section:
            if section.get(key):
                return False

        return True




    def list_data(self):
        """
        lists the data in the class by category
        returns type str of data
        """
        text = ""
        sections = [self.private,
                    self.protected,
                    self.public]
        section_names = ["private:\n",
                         "protected:\n",
                         "public:\n"]

        for i, section in enumerate(sections):
            text += section_names[i] + "\tstatic variables\n"
            for static_var in section.get("static_var"):
                text += "\t\t" + static_var.var_type + " " + static_var.name + "\n"

            text += "\tvariables\n"
            for var in section.get("var"):
                text += "\t\t" + var.var_type + " " + var.name + "\n"

            text += "\tstatic functions\n"
            for static_func in section.get("static_func"):
                text += "\t\t" + static_func.type + " " + static_func.name + "\n"

            text += "\tfunctions\n"
            for func in section.get("func"):
                text += "\t\t" + func.type + " " + func.name + "\n"

        return text




    def get_children(self):
        """
        returns a dict of children names and their object
        ex. {name1:obj1,
             name2:obj2,
             name3:obj3,
             ...}
        """
        children = {}
        sections = [self.private, self.protected, self.public]

        for section in sections:
            for static_var in section.get("static_var"):
                children.update({static_var.name:static_var})

            for var in section.get("var"):
                children.update({var.name:var})

            for static_func in section.get("static_func"):
                children.update({static_func.name:static_func})

```

```
114:             for func in section.get("func"):
115:                 children.update({func.name:func})
116: 
117:         return children
118: 
119: 
120:     def gen_header_text(self):
121:         """
122:         takes data in the class and generates text for a class in
123:         a header file
124:         returns type str of text
125:         """
126:         text = "class " + self.name + "\n{\n\tprivate:\n"
127: 
128:         #TODO: condense, function too long, too many branches
129:         for member in self.private.get("static_func"):
130:             text += "\t\t" + member.gen_header_text()
131:         if self.private.get("static_func"):
132:             text += "\n\n"
133: 
134:         for member in self.private.get("func"):
135:             text += "\t\t" + member.gen_header_text()
136:         if self.private.get("func"):
137:             text += "\n\n"
138: 
139:         for member in self.private.get("static_var"):
140:             text += "\t\t" + member.gen_header_text()
141:         if self.private.get("static_var"):
142:             text += "\n\n"
143: 
144:         for member in self.private.get("var"):
145:             text += "\t\t" + member.gen_header_text()
146:         if self.private.get("var"):
147:             text += "\n\n"
148: 
149: 
150:         prot_txt = ""
151:         if not self.__keys_empty(self.protected):
152:             #if not empty
153:             prot_txt = "\tprotected:\n"
154:             for member in self.protected.get("static_func"):
155:                 prot_txt += "\t\t" + member.gen_header_text()
156:             if self.protected.get("static_func"):
157:                 prot_txt += "\n\n"
158: 
159:             for member in self.protected.get("func"):
160:                 prot_txt += "\t\t" + member.gen_header_text()
161:             if self.protected.get("func"):
162:                 prot_txt += "\n\n"
163: 
164:             for member in self.protected.get("static_var"):
165:                 prot_txt += "\t\t" + member.gen_header_text()
166:             if self.protected.get("static_var"):
167:                 prot_txt += "\n\n"
168: 
169:             for member in self.protected.get("var"):
170:                 prot_txt += "\t\t" + member.gen_header_text()
171:             if self.protected.get("var"):
172:                 prot_txt += "\n\n"
173:         elif self.__keys_empty(self.protected) and not config.REMOVE_PROTECTED_IF_EMPTY:
174:             #if empty and dont remove protected
175:             prot_txt = "\tprotected:\n"
176: 
177:         text += prot_txt + "\tpublic:\n\t\t" + self.name + "();\n\t\t~" + self.name + "();\n\n"
178: 
179: 
180:         for member in self.public.get("static_func"):
181:             text += "\t\t" + member.gen_header_text()
182:         if self.public.get("static_func"):
183:             text += "\n\n"
184: 
185:         for member in self.public.get("func"):
186:             text += "\t\t" + member.gen_header_text()
187:         if self.public.get("func"):
188:             text += "\n\n"
189: 
190:         for member in self.public.get("static_var"):
191:             text += "\t\t" + member.gen_header_text()
192:         if self.public.get("static_var"):
193:             text += "\n\n"
194: 
195:         for member in self.public.get("var"):
196:             text += "\t\t" + member.gen_header_text()
197:         if self.public.get("var"):
198:             text += "\n\n"
199: 
200:         text += "};\n"
201: 
202:         tab = ""
203:         for _ in range(config.TAB_SIZE):
204:             tab += " "
205:         text = text.replace("\t", tab)
206: 
207:         return text
208: 
209: 
210: 
211: 
212:     def gen_impl_text(self):
213:         """
214:         generates text for a class in an implementation file
215:         returns type str of text
216:         """
217:         text = ""
218: 
219:         sections = [self.private, self.protected, self.public]
220:         for section in sections:
221:             for static_var in section.get("static_var"):
222:                 text += static_var.gen_impl_text()
223: 
224:         text += "\n\n\n" + self.name + "::" + self.name + "( )\n{\n\n}\n"
225:         text += "\n\n\n" + self.name + "::~" + self.name + "( )\n{\n\n}\n\n\n\n\n"
226: 
```

```
227:         for section in sections:
228:             for static_func in section.get("static_func"):
229:                 text += static_func.gen_impl_text() + "\n\n\n"
230:
231:             for func in section.get("func"):
232:                 text += func.gen_impl_text() + "\n\n\n"
233:
234:         return text
235:
236:
237:
238:
239: class cpp_func:
240:     """
241:     contains data about a cpp function type
242:     can be either static or not and has methods to
243:     generate text in both header and implementation file
244:     """
245:     def __init__(self, name, return_type, static, parent):
246:         self.type = return_type
247:         self.name = name
248:         self.static = bool(static)
249:
250:         self.has_children = False
251:         self.parent = parent
252:
253:         self.params = []
254:
255:
256:
257:
258:     def get_children(self):
259:         """
260:         returns a dict of children names and their object
261:         ex. {name1:obj1,
262:              name2:obj2,
263:              name3:obj3,
264:              ...}
265:         since there are no accesible children, an empty
266:         list is returned
267:         """
268:         return {}
269:
270:
271:
272:     def list_data(self):
273:         """
274:         lists the data in the function by category
275:         returns type str of data
276:         """
277:         text = "type:\n\t" + self.type + "\nname:\n\t" + self.name + "\nstatic:\n\t"
278:         text += "yes\n" if self.static else "no\n"
279:         text += "parameters:\n"
280:         for param in self.params:
281:             text += "\t" + param + "\n"
282:
283:         return text
284:
285:
286:
287:
288:     def gen_header_text(self):
289:         """
290:         generates text for a function with given dataset to be placed
291:         in a header file
292:         returns type str of text for header file
293:         """
294:         text = ""
295:         if self.static:
296:             text += "static "
297:
298:         text += self.type + " "
299:         text += self.name + "( "
300:         for param in self.params:
301:             text += param + ", "
302:         k = text.rfind(",")
303:         if k > 0:
304:             text = text[:k] + text[k+1:]
305:         text += ");\n"
306:
307:         return text
308:
309:
310:
311:
312:     def gen_impl_text(self, class_name=""):
313:         """
314:         generates text for a function with a given dataset to be placed
315:         in an implementation file
316:         returns type str of text for implementation file
317:         """
318:         text = ""
319:
320:         text += self.type + " "
321:         if class_name:
322:             text += class_name + "::" + self.name + "( "
323:         else:
324:             text += self.name + "( "
325:
326:         for param in self.params:
327:             text += param + ", "
328:         k = text.rfind(",")
329:         if k > 0:
330:             text = text[:k] + text[k+1:]
331:
332:         text += ")\n{\n\n}"
333:
334:         return text
335:
336:
337:
338:
339: class cpp_variable:
```

```
340:     """
341:     contains data about a cpp variable type
342:     can be either static or not and has methods to
343:     generate text in both header and implementation file
344:     """
345:     def __init__(self, name, var_type, static, parent):
346:         self.var_type = var_type
347:         self.name = name
348:         self.static = bool(static)
349:
350:         self.has_children = False
351:         self.parent = parent
352:
353:
354:
355:     def get_children(self):
356:         """
357:         returns a dict of children names and their object
358:         ex. {name1:obj1,
359:              name2:obj2,
360:              name3:obj3,
361:              ...}
362:         since there are no accesible children, an empty
363:         list is returned
364:         """
365:         return {}
366:
367:
368:
369:
370:     def list_data(self):
371:         """
372:         lists the data in the variable by category
373:         returns type str of data
374:         """
375:         text = "type:\n\t" + self.var_type + "\nname:\n\t" + self.name + "\nstatic:\n\t"
376:         text += "yes\n" if self.static else "no\n"
377:
378:         return text
379:
380:
381:
382:
383:     def gen_header_text(self):
384:         """
385:         generates text for a variable with given dataset to be placed
386:         in a header file
387:         returns type str of text for header file
388:         """
389:         text = "static " if self.static else ""
390:
391:         text += self.var_type + " " + self.name + ";"
392:
393:         return text
394:
395:
396:
397:
398:     def gen_impl_text(self):
399:         """
400:         generates text for a variable with a given dataset to be placed
401:         in an implementation file
402:         returns type str of text for implementation file
403:         """
404:         return self.var_type + " " + self.name + " = ;\n"
405:
406:
407:
408:
409:
410: class HeaderFile:
411:     """
412:     contains data for a cpp header file
413:     data can also be used to generate an implementation file
414:     """
415:     def __init__(self, file_name):
416:         self.file_name = file_name
417:
418:         self.parent = self
419:         self.has_children = True
420:
421:         self.user_includes = []
422:         self.lib_includes = []
423:         self.classes = []
424:         self.static_vars = []
425:         self.funcs = []
426:
427:
428:
429:
430:     def list_data(self):
431:         """
432:         lists the data in the header file by category
433:         returns type str of data
434:         """
435:         text = "file name:\n\t" + self.file_name + "\nlibrary includes:\n"
436:         for include in self.lib_includes:
437:             text += "\t" + include + "\n"
438:
439:         text += "user includes:\n"
440:         for include in self.user_includes:
441:             text += "\t" + include + "\n"
442:
443:         text += "classes:\n"
444:         for cpp_cls in self.classes:
445:             text += "\t" + cpp_cls.name + "\n"
446:
447:         text += "variables:\n"
448:         for static_var in self.static_vars:
449:             text += "\t" + static_var.gen_header_text() + "\n"
450:
451:         text += "functions:\n"
452:         for func in self.funcs:
```

```
453:            text += "\t" + func.type + " " + func.name + "\n"
454:
455:        return text
456:
457:
458:
459:
460:
461:    def get_children(self):
462:        """
463:        returns a dict of children names and their object
464:        ex. {name1:obj1,
465:             name2:obj2,
466:             name3:obj3,
467:             ...}
468:        """
469:        children = {}
470:
471:        for cpp_cls in self.classes:
472:            children.update({cpp_cls.name:cpp_cls})
473:
474:        for static_var in self.static_vars:
475:            children.update({static_var.name:static_var})
476:
477:        for func in self.funcs:
478:            children.update({func.name:func})
479:
480:        return children
481:
482:
483:
484:
485:    def gen_header_text(self):
486:        """
487:        generates text for a header file
488:        returns type str of text
489:        """
490:        #add header guards
491:        guard, _ = os.path.splitext(self.file_name)
492:        guard = guard.split("/")[-1]
493:
494:        text = "#ifndef __" + guard.upper() + "_HPP__\n"
495:        text += "#define __" + guard.upper() + "_HPP__\n\n"
496:
497:        for include in sorted(self.lib_includes):
498:            text += "#include <" + include + ">\n"
499:
500:        text += '\n#include "main.h"\n\n'
501:
502:        for include in sorted(self.user_includes):
503:            text += '#include "' + include + '"\n'
504:
505:        text += "\n\n"
506:
507:        for c in self.classes:
508:            text += c.gen_header_text() + "\n\n\n\n"
509:
510:        for func in self.funcs:
511:            text += func.gen_header_text() + "\n"
512:        if self.funcs:
513:            text += "\n\n\n\n"
514:
515:        for var in self.static_vars:
516:            text += var.gen_header_text()
517:        if self.static_vars:
518:            text += "\n\n\n\n"
519:
520:        text += "#endif\n"
521:
522:        return text
523:
524:
525:
526:
527:    def gen_impl_text(self):
528:        """
529:        generates text for an implementation file
530:        returns type str of text
531:        """
532:        text = ""
533:        for include in sorted(self.lib_includes):
534:            text += "#include <" + include + ">\n"
535:
536:        text += '\n#include "main.h"\n\n'
537:
538:        text += '#include "' + self.file_name + '"\n'
539:
540:        for include in sorted(self.user_includes):
541:            text += '#include "' + include + '"\n'
542:
543:        text += "\n\n"
544:
545:        for var in self.static_vars:
546:            text += var.gen_impl_text()
547:        if self.static_vars:
548:            text += "\n\n\n\n"
549:
550:        for c in self.classes:
551:            text += c.gen_impl_text() + "\n\n\n\n"
552:
553:        for func in self.funcs:
554:            text += func.gen_impl_text()
555:        if self.funcs:
556:            text += "\n\n\n\n"
557:
558:        return text
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Aug 20 16:48:22 2019

@author: aiden
"""

class Exit(Exception):
    """
    thrown for graceful program termination
    stack trace might not be shown if raised
    """
    pass

class UnknownOption(Exception):
    """
    thrown when an invalid parameter or and invalid command is given
    """
    def __init__(self):
        msg = "Unknown option given"
        super().__init__(msg)
        print(msg)

class InvalidFileName(Exception):
    """
    exception for and invalid file name given
    """
    def __init__(self):
        msg = ("Invalid file name given:\n"
               + "must end in valid header extension type ex.'.hpp' or '.h'")

        super().__init__(msg)
        print(msg)
        raise Exit

class InvalidAddition(Exception):
    """
    addition of object failed
    """
    def __init__(self, msg=None):
        if not msg:
            msg = ("adding of object failed: check to see that parent is of a valid type\n"
                   + "and that parameters were passed correctly")
        super().__init__(msg)
        print(msg)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Aug 17 19:53:35 2019

@author: aiden
"""
import colorama

TAB_SIZE = 4
REMOVE_PROTECTED_IF_EMPTY = True
SHELL_COLOR = colorama.Fore.LIGHTGREEN_EX
CURSOR_UP_ONE = '\x1b[1A'
ERASE_LINE = '\x1b[2K'
```

```
1: #!/bin/bash
2: #move this to /bin and give executable permissions
3: #to be able to run python script from anywhere
4: python3 /home/aiden/Documents/VexCode-2019/CodeGen/code_gen.py
```

```
1:  import tkinter as tk
2:
3:  import robot
4:  import field
5:  import fieldObjects
6:  import autonomous
7:  import runningFrame
8:  import robotInfoFrame
9:  import controlPanelFrame
10:
11:
12: #user defines
13: X_RES = 1600
14: Y_RES = 900
15: ROBOT_START_TILE = [0, 1]
16: ROBOT_START = [0, 6]
17: ROBOT_START_ANGLE = 0
18:
19:
20:
21: #sets up and draws field
22: f = field.Field(X_RES, Y_RES)
23: f.drawField()
24: f.drawGrid()
25: f.fieldInfo()
26: field = f.field
27: fieldSize = f.distance
28: robotCoordinates = field[ROBOT_START_TILE[0]][ROBOT_START_TILE[1]].placeRobot([ROBOT_START[0], ROBOT_START[1]])
29:
30: fieldObjects.main(field)
31:
32:
33:
34: #gets canvas and tkinter object
35: canvas = f.canvas
36: master = f.master
37:
38:
39:
40: #sets up GUI further
41: master.title("Autonomous Simulator")
42: master.wm_title("Autonomous Simulator")
43: icon = tk.Image("photo", file="autonSimulator.png")
44: # master.tk.call('wm', 'iconphoto', master._w, icon)
45: master.resizable(0, 0)
46:
47:
48: #sets up buttons and labels
49: runningPane = runningFrame.runningFrame(master)
50: robotInfoPane = robotInfoFrame.robotInfoFrame(master)
51: controlPanelPane = controlPanelFrame.controlPanelFrame(master, runningPane)
52:
53: runningPane.placeObjects()
54: robotInfoPane.placeObjects()
55: controlPanelPane.placeObjects()
56:
57:
58:
59: #moves robot to start location
60: bot = robot.robot(fieldSize=fieldSize, tkobj=master, canvas=canvas, controlPanelFrame=controlPanelPane, robotInfoFrame=robotInfoPane)
61: bot.show(angle=ROBOT_START_ANGLE, position=robotCoordinates)
62:
63:
64: def motion(event):
65:     x, y = event.x, event.y
66:     screen_offset_x = (X_RES - (Y_RES - 100)) / 2
67:     screen_offset_y = 50
68:     offset_x = bot.x_offset_in
69:     offset_y = bot.y_offset_in
70:     # print(bot.inches(x), bot.x_offset_in)
71:     corrected_coords = [bot.inches(x) - bot.y_offset_in, bot.inches(y) - bot.x_offset_in]
72:     print('{}, {}'.format(corrected_coords[0], corrected_coords[1]))
73:
74: def forward(event):
75:     bot.forward(25)
76:
77: def backward(event):
78:     bot.backward(25)
79:
80: def turnRight(event):
81:     bot.turnRight(2)
82:
83: def turnLeft(event):
84:     bot.turnLeft(2)
85:
86:
87: #runs autonomous
88: try:
89:     master.bind('<Motion>', motion)
90:     master.bind("w", forward)
91:     master.bind("s", backward)
92:     master.bind("a", turnLeft)
93:     master.bind("d", turnRight)
94:
95:     auton = autonomous.auton(bot, master, controlPanelPane, robotInfoPane)
96:     auton.commands()
97:
98: except tk.TclError:
99:     pass
100:
101: finally:
102:     try: #finished all the way through
103:         controlPanelPane.disable()
104:
105:         controlPanelPane.idle()
106:
107:     except tk.TclError: #window closed
108:         print("finished")
109:
110:
111:
112:
113:
```

114:
115:
116:
117:
118:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import time

class auton:
    """
    contains autonomous commands
    """
    def __init__(self, robotObj, master, controlPanelFrame, robotInfoFrame):
        self.controlPanelFrame = controlPanelFrame
        self.robotInfoFrame = robotInfoFrame
        self.robot = robotObj
        self.master = master

        self.orientationLabelText = self.robotInfoFrame.orientationLabelText
        self.orientationLabelText.set("orientation: " + str(self.robot.orientationDegrees))

        self.controlPanelFrame.runningLabelText.set(self.controlPanelFrame.options.get(str(self.controlPanelFrame.keepRunning)))



    def nextFrame(self, waitTime=0):
        """
        updates tkinter canvas and allows for a wait
        """

        self.master.update()

        timeSlept = 0
        while timeSlept <= waitTime:
            if self.controlPanelFrame.keepRunning:
                time.sleep(0.1)
                timeSlept += 0.1
                self.master.update()
            else: #sets robot to idle
                time.sleep(0.1)
                self.master.update()


    def intakeStart(self, rotations):
        """
        for translator
        """
        pass

    def outakeStart(self, rotations):
        """
        for translator
        """
        pass

    def intakeEnd(self, Time):
        """
        for translator
        """
        pass

    def catapult(self, rotations):
        """
        for translator
        """
        pass

    def capFlipper(self, rotations):
        """
        for translator
        """
        pass


    def commands(self):
        """
        autonomous commands
        """
        self.nextFrame(1)

        self.robot.drive_to_point(0, 27)
        self.nextFrame(.25)

        self.robot.drive_to_point(27.2, 5.5, 1)
        self.nextFrame(.25)

        self.robot.drive_to_point(0, 27)
        self.nextFrame(.25)

        self.robot.drive_to_point(-22.5, 4.8, 1)
        self.nextFrame(.25)

        # self.robot.forward(350)
        # self.nextFrame(.25)

        #self.robot.turnLeft(10)
        #self.nextFrame(.25)


        # self.robot.drive_to_point(0, 30)
        # self.nextFrame(.25)


        # self.robot.drive_to_point(0, 0)
        # self.nextFrame(1)




########################## unit test ##########################
#       import random                           #
#       for x in range(0,4):                    #
#           self.robot.forward(500)               #
#           self.nextFrame(1)                   #
```

```
114: #                                  #
115: #          self.robot.turnLeft(90)          #
116: #          self.nextFrame(1)                #
117: #                                  #
118: #      self.robot.turnLeft(45)              #
119: #      self.nextFrame(1)                    #
120: #                                  #
121: #      for x in range(0,4):                 #
122: #          self.robot.forward(400)          #
123: #          self.nextFrame(1)                #
124: #                                  #
125: #          self.robot.turnRight(90)         #
126: #          self.nextFrame(1)                #
127: #                                  #
128: #      self.robot.turnRight(45)             #
129: #      self.nextFrame(1)                    #
130: #                                  #
131: #                                  #
132: #      for x in range(0, 6):                #
133: #          self.robot.forward(500)          #
134: #          self.nextFrame(1)                #
135: #                                  #
136: #          self.robot.turnRight(random.randint(10,90))    #
137: #          self.nextFrame(1)                #
138: #                                  #
139: ####################################################################
140:
141:
142:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import smtplib
import getpass


SEND_EMAIL = 0
FILE = 'autonomous.py'


class email():
    def __init__(self):
        self.passwd = None
        self.user_email = 'jg570144@gmail.com'
        self.recipients = []

        self.message = None

        self.smtpObj = None
        self.check = None
        self.subject = None
        self.body = None
        self.sent = 0


    def getCredentials(self):
        #self.user_email = input("email address:\n")
        self.passwd = getpass.getpass()


    def getMessage(self, oldCode, newCode):
        r = int(input("\nEnter number of recipients\n"))

        for x in range(r):
            remail = input("enter recipient email\n")
            self.recipients.append(remail)

        self.subject = "translated autonomous code"

        self.body = "old auton code:\n\n"

        for i in oldCode:
            self.body = self.body + i + '\n'
            print("")

        self.body = self.body + '\n\n\n\nnew auton code:\n\n'

        for i in finalCode:
            self.body = self.body + i + '\n'

        self.body = self.body + '\n\n\nthis is an automated message\ncourtesy of Aiden'
        print("")


    def login(self):
        self.smtpObj = smtplib.SMTP("smtp.gmail.com:587")
        self.smtpObj.starttls()
        self.smtpObj.ehlo()
        self.smtpObj.login(self.user_email, self.passwd)


    def send(self):
        print("sent: ")

        message = "Subject: " + self.subject + "\n" + self.body

        self.check = self.smtpObj.sendmail(self.user_email, self.recipients, message)

    def close(self):
        self.smtpObj.quit()









class Translator:

    def __init__(self, file):
        self.file = open(file)
        self.code = self.file.readlines()

        self.oldCode = []
        self.newCode = []

        self.conversion = {
            "turnRight": "turnRightV",
            "turnLeft": "turnLeftV",
            "rightSide": "rs",
            "leftSide": "ls",
            "forward": "driveForward",
            "backward": "driveForward",
            "reverse": "changeDirection",
            "intakeStart": "intakeStart",
            "intakeEnd": "intakeEnd",
            "outake": "intakeStart",
            "catapult": "shootBall",
            "capFlipper": "flip"
            }


        self.functionMap = {
            "value": self.value,
            "sleep100RPM": self.sleep100RPM,
```

```
            "sleep200RPM": self.sleep200RPM,
            "sleep600RPM": self.sleep600RPM,
        "negativeValue": self.negativeValue,
        "noParam": self.noParam
         }


    self.specialInstructions = {
        "forward": ["value", "sleep200RPM"],
        "backward": ["negativeValue", "sleep200RPM"],
        "turnLeft": ["noParam"],
        "turnRight": ["noParam"],
        "intakeStart": ["negativeValue"],
        "outakeStart": ["value"],
        "intakeEnd": ["sleep600"],
        "catapult": ["value", "sleep100"],
        "capFlipper": ["value"],
        }



    self.parameterValue = ""
    self.sleepTime = ""
    self.separator = ""
    self.instructions = []


def value(self):
    self.parameterValue = str(self.parameterValue)
    self.sleepTime = ""
    self.separator = ""

def sleep100RPM(self):
    RPM = 100
    RPS = RPM / 60
    revolutions = (abs(float(self.parameterValue)) / 360)
    self.sleepTime = (1000 * (revolutions / RPS)) + 50
    #self.sleepTime = str(int((1000 * abs((float(self.parameterValue))/360))) + 200)
    if len(self.instructions) == 1:
        self.parameterValue = ""
        self.separator = ""
    else:
        self.separator = ", "

def sleep200RPM(self):
    RPM = 200
    RPS = RPM / 60
    revolutions = (abs(float(self.parameterValue)) / 360)
    self.sleepTime = (1000 * (revolutions / RPS)) + 50
    if len(self.instructions) == 1:
        self.parameterValue = ""
        self.separator = ""
    else:
        self.separator = ", "

def sleep600RPM(self):
    RPM = 600
    RPS = RPM / 60
    revolutions = (abs(float(self.parameterValue)) / 360)
    self.sleepTime = (1000 * (revolutions / RPS)) + 50
    if len(self.instructions) == 1:
        self.parameterValue = ""
        self.separator = ""
    else:
        self.separator = ", "


def negativeValue(self):
    self.parameterValue = str(0 - float(self.parameterValue))
    self.sleepTime = ""
    self.separator = ""

def noParam(self):
    self.parameterValue = ""
    self.sleepTime = ""
    self.separator = ""




def translate(self):

    for i in self.code:

        try:
            oldCommand = i.split(".")[1]
            try:
                rawCommand = i.split(".")[2]
                oldCommand = oldCommand + "." + rawCommand
            except:
                rawCommand = i.split(".")[1]

            oldCommand = oldCommand.split(")")[0]
            oldCommand = 'self.' + oldCommand + ")"

            command = rawCommand.split("(")[0]
            value = rawCommand.split("(")[1]
            self.parameterValue = value.split(")")[0]

            self.instructions = list(self.specialInstructions.get(command))
            for j in self.instructions:
                self.functionMap[str(j)]()

            newCommand = self.conversion.get(command)
            newCommand = (newCommand
                    + "(" + str(self.parameterValue)
                    + str(self.separator)
                    + str(self.sleepTime)
                    + ");"
                    )

            self.oldCode.append(oldCommand)
```

```
227:                self.newCode.append(newCommand)
228:                self.parameterValue = ""
229:                self.sleep = ""
230:                self.separator = ""
231:
232:            except:
233:                self.parameterValue = ""
234:                self.sleep = ""
235:                self.separator = ""
236:
237:        return self.oldCode, self.newCode
238:
239:
240:
241:  t = Translator(FILE)
242:  oldCode, finalCode = t.translate()
243:
244:
245:  if SEND_EMAIL:
246:      s = email()
247:      s.getCredentials()
248:      s.getMessage(oldCode, finalCode)
249:      s.login()
250:      s.send()
251:      s.close()
252:
253:      print("")
254:      print("")
255:      print("done")
256:
257:  else:
258:      for i in oldCode:
259:          print(i)
260:      print("")
261:      for i in finalCode:
262:          print(i)
263:
264:
265:
266:
267:
268:
269:
270:
271:
272:
273:
274:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import tkinter as tk
import time

class controlPanelFrame:
    """
    makes frame object for all labels and buttons
    """
    def __init__(self, master, runningFrame):
        self.runningLabelText = runningFrame.runningLabelText
        self.master = master

        self.guiFrame = tk.Frame(self.master)
        self.guiFrame.grid(row=1, column=0, sticky='news')

        self.speed = 7
        self.keepRunning = True

        self.options = {
            "True": "running",
            "False":"paused "
            }

        self.pauseButtonTextOptions = {
            "True": "pause ",
            "False":"resume"
             }


        ##### labels #####

        #speed label
        self.speedLabelText = tk.StringVar()
        self.speedLabel = tk.Label(self.guiFrame, textvariable=self.speedLabelText)


        ##### buttons #####

        self.fasterButton = tk.Button(self.guiFrame, text=' + ', command=self.__faster)
        self.slowerButton = tk.Button(self.guiFrame, text=' - ', command=self.__slower)

        self.pauseButtonText = tk.StringVar()
        self.pauseButton = tk.Button(self.guiFrame, textvariable=self.pauseButtonText, command=self.__pause)
        self.master.bind("<space>", self.__pause)




    def __changeSpeedLabel(self):
        self.speedLabelText.set("speed: " + str(self.speed))



    def __faster(self):
        if self.speed < 10:
            self.speed = self.speed + 1
            self.__changeSpeedLabel()

    def __slower(self):
        if self.speed > 1:
            self.speed = self.speed - 1
            self.__changeSpeedLabel()

    def __pause(self, event=None):
        """
        pauses robot actions
        """
        self.keepRunning = not self.keepRunning

        self.runningLabelText.set(self.options.get(str(self.keepRunning)))
        self.pauseButtonText.set(self.pauseButtonTextOptions.get(str(self.keepRunning)))

        self.master.update()


    def idle(self):
        """
        sends canvas into idle state
        """
        while 1:
            time.sleep(0.1)
            self.master.update()

    def disable(self):
        self.pauseButton.config(state="disabled")
        self.fasterButton.config(state="disabled")
        self.slowerButton.config(state="disabled")

        self.master.unbind("<space>")

    def placeObjects(self):
        """
        places all buttons and labels
        """

        self.pauseButton.grid(row=0, column=2, columnspan=4, rowspan=2, sticky='news', ipadx=30)

        self.speedLabel.grid(row=2, column=2, columnspan=4, rowspan=1, sticky='news')

        self.slowerButton.grid(row=0, column=0, rowspan=3, columnspan=2, sticky='news')
        self.fasterButton.grid(row=0, column=6, rowspan=3, columnspan=2, sticky='news')


        #sets default value of labels
        self.pauseButtonText.set(self.pauseButtonTextOptions.get(str(self.keepRunning)))
        self.speedLabelText.set("speed: " + str(self.speed))
        self.runningLabelText.set(self.options.get(str(self.keepRunning)))
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-


def main(field):
    """
    creates field elements
    """
    for num in range(0,6): #draw white line auton line
        field[2][num].drawRectangle(vertexes=[[47,0],[47,47],[47,47],[47,0]], color="#ffffff", width=10, outline="#ffffff")

    for num in range(0,6): #draw red alliance starting line
        field[4][num].drawRectangle(vertexes=[[47,0],[47,47],[47,47],[47,0]], color="#ffffff", width=7, outline="#ffffff")

    for num in range(0,6): #draw blue alliance starting line
        field[0][num].drawRectangle(vertexes=[[47,0],[47,47],[47,47],[47,0]], color="#ffffff", width=7, outline="#ffffff")

    field[0][3].drawRectangle(vertexes=[[0,0],[47,0],[47,0],[0,0]], color="#ffffff", width=10, outline="#ffffff")
    field[5][3].drawRectangle(vertexes=[[0,0],[47,0],[47,0],[0,0]], color="#ffffff", width=10, outline="#ffffff")

    #corner goal
    field[0][0].drawObjectFieldElementCircle(position=[11, 11], color='#919191', size=10)
    field[5][0].drawObjectFieldElementCircle(position=[36, 10], color='#919191', size=10)
    field[0][5].drawObjectFieldElementCircle(position=[11, 38], color='#919191', size=10)
    field[5][5].drawObjectFieldElementCircle(position=[36, 38], color='#919191', size=10)

    #cross goals
    field[3][0].drawObjectFieldElementCircle(position=[0, 10], color='#919191', size=10)
    field[3][3].drawObjectFieldElementCircle(position=[0, 0], color='#919191', size=10)
    field[3][5].drawObjectFieldElementCircle(position=[0, 38], color='#919191', size=10)

    field[0][3].drawObjectFieldElementCircle(position=[10, 0], color='#919191', size=10)
    field[5][3].drawObjectFieldElementCircle(position=[36, 0], color='#919191', size=10)


    #game objects

    #corner balls
    field[0][0].drawObjectFieldElementCircle(position=[24, 24], color='blue', size=7)
    field[0][5].drawObjectFieldElementCircle(position=[24, 24], color='blue', size=7)

    field[5][0].drawObjectFieldElementCircle(position=[24, 24], color='red', size=7)
    field[5][5].drawObjectFieldElementCircle(position=[24, 24], color='red', size=7)

    #middle balls
    field[3][2].drawObjectFieldElementCircle(position=[0, 29], color='blue', size=7)
    field[3][3].drawObjectFieldElementCircle(position=[18, 0], color='blue', size=7)

    field[3][3].drawObjectFieldElementCircle(position=[0, 18], color='red', size=7)
    field[2][3].drawObjectFieldElementCircle(position=[29, 0], color='red', size=7)

    #other balls
    field[3][1].drawObjectFieldElementCircle(position=[0, 24], color='blue', size=7)
    field[3][4].drawObjectFieldElementCircle(position=[0, 24], color='red', size=7)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import tkinter as tk


class Tile:
    """
    gives ability to draw objects on each tile
    """
    def __init__(self, x, y, distance, canvas):
        self.canvas = canvas

        self.distance = distance

        self.P1 = [x, y]
        self.P2 = [x + distance, y]
        self.P3 = [x, y + distance]
        self.P4 = [x + distance, y + distance]


        self.grid = []
        x = self.P1[0]
        increment = self.distance/47
        for i in range(0, 48): #makes grid with coordinates
            column = []
            y = self.P1[1]
            for j in range(0, 48):
                coords = [x, y]
                column.append(coords)
                y = y + increment
            x = x + increment
            self.grid.append(column)


    def __makeCenteredGrid(self, size):
        centeredGrid = []
        x = self.P1[0]
        increment = (self.distance - size) / 2
        for i in range(0, 3): #makes grid with coordinates
            column = []
            y = self.P1[1]
            for j in range(0, 3):
                coords = [x, y]
                column.append(coords)
                y = y + increment
            x = x + increment
            centeredGrid.append(column)

        return centeredGrid


    def drawObjectTiles(self, position=[23, 23], size=24, color="#cccccc", outline=None, width=None):
        """
        draws object to fill a square of the grid in inches
        """
        x1 = self.grid[position[0]][position[1]][0]
        y1 = self.grid[position[0]][position[1]][1]

        s = self.distance/25
        for i in range(0, size):
            s = s + (self.distance/25)

        x1 = x1 - (s/2)
        y1 = y1 - (s/2)

        x2 = self.grid[position[0]][position[1]][0] + (s/2)
        y2 = self.grid[position[0]][position[1]][1] + (s/2)

        self.canvas.create_rectangle(x1, y1, x2, y2, outline=outline, fill=color, width=width)


    def drawObjectFieldElementSquare(self, position=[1, 1], size=8, color="#cccccc", outline=None, width=None):
        """
        draws object centered on the coordinate chosen in inches
        """
        x1 = self.grid[position[0]][position[1]][0]
        y1 = self.grid[position[0]][position[1]][1]

        s = self.distance/25
        for i in range(0, size):
            s = s + (self.distance/25)

        x1 = x1 - (s/2)
        y1 = y1 - (s/2)

        x2 = self.grid[position[0]][position[1]][0] + (s/2)
        y2 = self.grid[position[0]][position[1]][1] + (s/2)

        self.canvas.create_rectangle(x1, y1, x2, y2, outline=outline, fill=color, width=width)


    def drawObjectFieldElementCircle(self, position=[1, 1], size=8, color="#cccccc", outline=None, width=None):
        """
        draws object centered on the coordinate chosen in inches
        """
        x1 = self.grid[position[0]][position[1]][0]
        y1 = self.grid[position[0]][position[1]][1]

        s = self.distance/25
        for i in range(0, size):
            s = s + (self.distance/25)

        x1 = x1 - (s/2)
        y1 = y1 - (s/2)

        x2 = self.grid[position[0]][position[1]][0] + (s/2)
        y2 = self.grid[position[0]][position[1]][1] + (s/2)

        self.canvas.create_oval(x1, y1, x2, y2, outline=outline, fill=color, width=width)


```

```
114:
115:
116:    def drawObjectFieldElementCentered(self, position=[1, 1], size=8, color="#cccccc", outline=None, width=None):
117:        """
118:        draws object contained only in tile in inches
119:        """
120:        s = self.distance/25
121:        for i in range(0, size):
122:            s = s + (self.distance/25)
123:
124:        centeredGrid = self.__makeCenteredGrid(s)
125:
126:        x1 = centeredGrid[position[0]][position[0]][0]
127:        y1 = centeredGrid[position[0]][position[1]][1]
128:
129:        x2 = centeredGrid[position[0]][position[0]][0] + s
130:        y2 = centeredGrid[position[0]][position[1]][1] + s
131:
132:        self.canvas.create_rectangle(x1, y1, x2, y2, outline=outline, fill=color, width=width)
133:
134:
135:    def drawRectangle(self, vertexes=[[0, 1], [2, 1], [2, 1], [0, 1]], color="#cccccc", outline="white", width=3):
136:        """
137:        draws a rectangle given four vertices
138:        """
139:        points = [
140:            [
141:                self.grid[vertexes[0][0]][vertexes[0][0]][0],
142:                self.grid[vertexes[0][1]][vertexes[0][1]][1]
143:            ],
144:            [
145:                self.grid[vertexes[1][0]][vertexes[1][0]][0],
146:                self.grid[vertexes[1][1]][vertexes[1][1]][1]
147:            ],
148:            [
149:                self.grid[vertexes[2][0]][vertexes[2][0]][0],
150:                self.grid[vertexes[2][1]][vertexes[2][1]][1]
151:            ],
152:            [
153:                self.grid[vertexes[3][0]][vertexes[3][0]][0],
154:                self.grid[vertexes[3][1]][vertexes[3][1]][1]
155:            ]
156:
157:            ]
158:
159:        self.canvas.create_polygon(points, outline=outline, fill=color, width=width)
160:
161:
162:
163:    def placeRobot(self, position=[1, 1], size=17):
164:        """
165:        draws robot in tile in inches
166:        """
167:        s = self.distance/25
168:        for i in range(0, size):
169:            s = s + (self.distance/25)
170:
171:        #centeredGrid = self.__makeCenteredGrid(s)
172:
173:        x1 = self.grid[position[0]][position[0]][0]
174:        y1 = self.grid[position[0]][position[1]][1]
175:
176:        x2 = self.grid[position[0]][position[0]][0] + s
177:        y2 = self.grid[position[0]][position[1]][1] + s
178:
179:        coords = [[x1, y1], [x2, y2]]
180:        return coords
181:
182:
183:
184:
185:
186:
187:
188:
189:  class Field:
190:      """
191:      creates the field
192:      """
193:      def __init__(self, width, height):
194:          master = tk.Tk()
195:          self.master = master
196:          self.canvas = tk.Canvas(master, width=width, height=height)
197:          self.canvas.grid(row=0, column=0, columnspan=3)
198:
199:          self.canvas.create_rectangle(0, 0, width, height, fill="#ffffff")
200:
201:          distance = height - 100
202:          self.P1 = [((width-distance)/2), 50]
203:          self.P2 = [(((width-distance)/2) + (distance)), 50]
204:          self.P3 = [((width-distance)/2), (50+distance)]
205:          self.P4 = [(((width-distance)/2) + (distance)), (50+distance)]
206:
207:          self.width = width
208:          self.height = height
209:          self.distance = distance
210:
211:          self.field = []
212:
213:
214:
215:
216:      def drawField(self):
217:          """
218:          draws the field based on the resolution given
219:          """
220:          self.canvas.create_rectangle(self.P1[0], self.P1[1], self.P4[0], self.P4[1], fill="#cccccc")
221:
222:          self.canvas.create_line(self.P1[0], self.P1[1], self.P2[0], self.P2[1], fill="black", width=7) #horizontal
223:          self.canvas.create_line(self.P1[0], self.P1[1] - 3, self.P3[0], self.P3[1] + 4, fill="black", width=7) #vertical
224:
225:          self.canvas.create_line(self.P2[0], self.P2[1] - 3, self.P4[0], self.P4[1] + 4, fill="black", width=7) #vertical
226:          self.canvas.create_line(self.P3[0], self.P3[1], self.P4[0], self.P4[1], fill="black", width=7) #horizontal
```

```
227:
228:
229:
230:     def drawGrid(self):
231:         """
232:         draws grid on the field
233:         """
234:         start = (self.width - self.distance) / 2
235:         increment = self.distance / 6
236:
237:         for x in range(0, 6): #verical lines
238:             P1 = [start + (increment*x), 50]
239:             P2 = [start + (increment*x), (50+self.distance)]
240:
241:             self.canvas.create_line(P1[0], P1[1], P2[0], P2[1], fill="black", width=2)
242:
243:         start = 50
244:         for x in range(0, 6): #horizontal lines
245:             P1 = [((self.width - self.distance) / 2), start + (increment*x)]
246:             P2 = [(((self.width - self.distance) / 2) + self.distance), start + (increment*x)]
247:
248:             self.canvas.create_line(P1[0], P1[1], P2[0], P2[1], fill="black", width=2)
249:
250:
251:     def fieldInfo(self):
252:         """
253:         creates objects of each tile and appends them to a list
254:         so that it is possible to create objects on the field
255:         """
256:         distance = self.distance/6
257:         x = ((self.width-self.distance)/2)
258:
259:         for i in range(0, 6): #columns
260:             column = []
261:             y = 50
262:             for j in range(0, 6): #rows
263:                 tileObject = Tile(x, y, distance, self.canvas)
264:                 column.append(tileObject)
265:
266:                 y = y + distance
267:             x = x + distance
268:
269:             self.field.append(column)
270:
271:
272:
273:
274:
275:
276:
277:
278:
279:
280:
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Jan  5 08:40:28 2021

@author: aiden
"""
import math
def to_radians(degrees):
    return ((degrees * math.pi) / 180)

def to_degrees(radians):
    return ((radians * 180) / math.pi)

current_x = -36
current_y = 0
current_angle = to_radians(-135)

end_x = 0
end_y = 0


dx = end_x - current_x
dy = end_y - current_y
print(to_degrees(math.atan2(dy, dx)))
dtheta = (math.atan2(dy, dx))
if dtheta < 0:
    dtheta += 2 * math.pi

if current_angle < 0:
    current_angle += 2 * math.pi

current_angle = -current_angle + (math.pi/2)
to_turn = current_angle - dtheta
print(to_degrees(current_angle), to_degrees(dtheta))
if to_turn > math.pi:
    to_turn = (-2 * math.pi) + to_turn
if to_turn < -math.pi:
    to_turn = (2 * math.pi) + to_turn

print(to_degrees(to_turn))
```

```
1: smtplib
2: getpass
3: tkinter
4:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import tkinter as tk

class robotInfoFrame:
    """
    makes frame object for all labels and buttons
    """
    def __init__(self, master):
        self.master = master

        self.guiFrame = tk.Frame(self.master)
        self.guiFrame.grid(row=1, column=2, sticky='news')


        ##### labels #####

        # position label
        self.positionLabelText = tk.StringVar()
        self.positionLabel = tk.Label(self.guiFrame, textvariable=self.positionLabelText)

        #orientation label
        self.orientationLabelText = tk.StringVar()
        self.orientationLabel = tk.Label(self.guiFrame, textvariable=self.orientationLabelText)

        #distance moved label
        self.distanceMovedLabelText = tk.StringVar()
        self.distanceMovedLabel = tk.Label(self.guiFrame, textvariable=self.distanceMovedLabelText)

        #current command label
        self.commandLabelText = tk.StringVar()
        self.commandLabel = tk.Label(self.guiFrame, textvariable=self.commandLabelText)

        #white space
        self.whiteSpaceLabel = tk.Label(self.guiFrame)

        #trailing white space
        self.trailingWhiteSpaceLabel = tk.Label(self.guiFrame)



    def placeObjects(self):
        """
        places all labels and sets default value
        """
        self.positionLabel.grid(row=0, column=8, columnspan=1, sticky='w')
        self.orientationLabel.grid(row=1, column=8, columnspan=1, sticky='w')
        self.distanceMovedLabel.grid(row=2, column=8, columnspan=1, sticky='w')
        self.commandLabel.grid(row=3, column=8, columnspan=1, sticky='w')

        self.whiteSpaceLabel.grid(row=0, column=0, columnspan=7, rowspan=3, sticky='news')
        self.trailingWhiteSpaceLabel.grid(row=0, column=4, sticky='news')


        #configures trailing white space to be eaten
        self.guiFrame.grid_columnconfigure(4, weight=1)

        self.positionLabelText.set("(x: ?, y: ?)")
        self.orientationLabelText.set("orientation: ")
        self.distanceMovedLabelText.set("distance moved: N/A                    ")
        self.commandLabelText.set("command: ")
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from PIL import Image
import io
import time
import math
import stopwatch


class robot:
    """
    contains all robot move functions
    """
    def __init__(self, fieldSize=None, tkobj=None, canvas=None, diameterOfWheel=4.1, controlPanelFrame=None, robotInfoFrame=None):
        self.controlPanelFrame = controlPanelFrame
        self.robotInfoFrame = robotInfoFrame

        self.canvas = canvas
        self.master = tkobj

        self.diameterOfWheel = diameterOfWheel
        self.fieldSize = fieldSize
        self.reversed = 0

        self.sqaure = None
        self.line = None

        self.squareVertexes = []
        self.lineVertexes = []

        self.iteration = 0

        self.x_offset_in = 0
        self.y_offset_in = 0
        self.theta_offset_deg = 0

    def __calcSleepTime(self, distance, iterations):
        return (distance / (10*(2 ** self.controlPanelFrame.speed))) / iterations


    def __generate_postscript(self):
        # ps = self.canvas.postscript(colormode="color")
        # im = Image.open(io.BytesIO(ps.encode('utf-8')))
        # im.save("ps/test" + str(self.iteration) + ".jpg")
        # self.iteration += 1
        pass

    def __calcCenters(self):
        """
        returns center of each polygon
        """
        xVals = []
        yVals = []

        for i in range(0, len(self.squareVertexes)): #calculates square center
            xVals.append(self.squareVertexes[i][0])
            yVals.append(self.squareVertexes[i][1])

        xVals.sort(reverse=True)
        yVals.sort(reverse=True)

        greatestX = xVals[0]
        greatestY = yVals[0]

        xVals.sort()
        yVals.sort()

        leastX = xVals[0]
        leastY = yVals[0]

        x = abs(((greatestX - leastX)/2)) + leastX
        y = abs(((greatestY - leastY)/2)) + leastY

        center = [x, y]


        return center




    def __rotate(self, Angle, pivotPoint):
        """
        rotates the robot simulating one side of the
        chassis moving
        """
        center = (pivotPoint[0], pivotPoint[1])
        angle = math.radians(Angle)
        cos_val = math.cos(angle)
        sin_val = math.sin(angle)
        cx, cy = center

        new_points = []
        for x_old, y_old in self.squareVertexes:
            x_old -= cx
            y_old -= cy
            x_new = x_old * cos_val - y_old * sin_val
            y_new = x_old * sin_val + y_old * cos_val
            new_points.append([x_new + cx, y_new + cy])


        angle = math.radians(Angle)
        cos_val = math.cos(angle)
        sin_val = math.sin(angle)
        cx, cy = center

        new_points2 = []
        for x_old, y_old in self.lineVertexes:
            x_old -= cx
            y_old -= cy
            x_new = x_old * cos_val - y_old * sin_val
            y_new = x_old * sin_val + y_old * cos_val
```

```
114:            new_points2.append([x_new + cx, y_new + cy])
115:
116:        self.squareVertexes = new_points
117:        self.lineVertexes = new_points2
118:
119:
120:
121:
122:    def __rotateInPlace(self, Angle):
123:        """
124:        rotates the robot in place simulating both sides of the chassis
125:        moving
126:        """
127:        center = self.__calcCenters()
128:
129:
130:        center = (center[0], center[1])
131:
132:        angle = math.radians(Angle) #moves square
133:        cos_val = math.cos(angle)
134:        sin_val = math.sin(angle)
135:        cx, cy = center
136:        new_points = []
137:        for x_old, y_old in self.squareVertexes:
138:            x_old -= cx
139:            y_old -= cy
140:            x_new = x_old * cos_val - y_old * sin_val
141:            y_new = x_old * sin_val + y_old * cos_val
142:            new_points.append([x_new + cx, y_new + cy])
143:
144:        angle = math.radians(Angle) #moves line
145:        cos_val = math.cos(angle)
146:        sin_val = math.sin(angle)
147:        cx, cy = center
148:        new_points2 = []
149:        for x_old, y_old in self.lineVertexes:
150:            x_old -= cx
151:            y_old -= cy
152:            x_new = x_old * cos_val - y_old * sin_val
153:            y_new = x_old * sin_val + y_old * cos_val
154:            new_points2.append([x_new + cx, y_new + cy])
155:
156:
157:        self.squareVertexes = new_points
158:        self.lineVertexes = new_points2
159:
160:
161:
162:    def __move(self, units):
163:        """
164:        simulates the robot moving in a straight line
165:        """
166:        quadrants = {  #quadrant: [xVal, yVal]
167:            1:[1, 1],
168:            2:[-1, 1],
169:            3:[-1, -1],
170:            4:[1, -1]
171:            }
172:
173:
174:        #determine quadrant of final position
175:        #used to determine if robot needs to add or subtract
176:        #x and y value to move in that direction
177:
178:        if units > 0 and (self.orientationDegrees >= 0 and self.orientationDegrees < 90):
179:            quadrant = 1
180:        elif units < 0 and (self.orientationDegrees >= 180 and self.orientationDegrees <= 270):
181:            quadrant = 1
182:
183:        elif units > 0 and (self.orientationDegrees >= 90 and self.orientationDegrees <= 180):
184:            quadrant = 2
185:        elif units < 0 and (self.orientationDegrees > 270 and self.orientationDegrees < 360):
186:            quadrant = 2
187:
188:        elif units > 0 and (self.orientationDegrees >= 180 and self.orientationDegrees < 270):
189:            quadrant = 3
190:        elif units < 0 and (self.orientationDegrees >= 0 and self.orientationDegrees <= 90):
191:            quadrant = 3
192:
193:        elif units > 0 and (self.orientationDegrees >= 270 and self.orientationDegrees < 360):
194:            quadrant = 4
195:        elif units < 0 and (self.orientationDegrees > 90 and self.orientationDegrees < 180):
196:            quadrant = 4
197:
198:
199:
200:        vals = quadrants.get(quadrant)
201:        xPol = vals[0]
202:        yPol = vals[1]
203:
204:        #absolute value simulates reference angle
205:        #trig functions of reference angles are positive
206:        x = xPol * abs(math.cos(math.radians(self.orientationDegrees)))
207:        y = yPol * abs(math.sin(math.radians(self.orientationDegrees)))
208:
209:        d = (math.sqrt((x**2) + (y**2)))
210:        distanceMoved = 0
211:
212:        stp = stopwatch.stopwatch()
213:        stp.start()
214:
215:        iterations = int(abs(units)/d)
216:        for i in range(0, iterations): #move animation
217:            if not self.controlPanelFrame.keepRunning: #allows for pause
218:                while not self.controlPanelFrame.keepRunning:
219:                    time.sleep(0.1)
220:                    self.master.update()
221:
222:            self.canvas.move(self.square, x, y)
223:            self.canvas.move(self.line, x, y)
224:
225:            self.master.update()
226:            self.__generate_postscript()
```

```
227:         time.sleep(self.__calcSleepTime(self.__encoderTicks(abs(units)), iterations))
228:
229:         eus = self.__encoderTicks(d)  # shows distance moved
230:         distanceMoved = distanceMoved + eus
231:
232:         #updates vertices of square and line
233:         xChange = (xPol * abs(math.cos(math.radians(self.orientationDegrees)) * abs(units))) / iterations
234:         yChange = (yPol * abs(math.sin(math.radians(self.orientationDegrees)) * abs(units))) / iterations
235:
236:         self.squareVertexes = [
237:             [self.squareVertexes[0][0] + xChange, self.squareVertexes[0][1] + yChange],
238:             [self.squareVertexes[1][0] + xChange, self.squareVertexes[1][1] + yChange],
239:             [self.squareVertexes[2][0] + xChange, self.squareVertexes[2][1] + yChange],
240:             [self.squareVertexes[3][0] + xChange, self.squareVertexes[3][1] + yChange]
241:             ]
242:
243:         self.lineVertexes = [
244:             [self.lineVertexes[0][0] + xChange, self.lineVertexes[0][1] + yChange],
245:             [self.lineVertexes[1][0] + xChange, self.lineVertexes[1][1] + yChange],
246:             [self.lineVertexes[2][0] + xChange, self.lineVertexes[2][1] + yChange],
247:             [self.lineVertexes[3][0] + xChange, self.lineVertexes[3][1] + yChange]
248:             ]
249:
250:         self.__update()
251:
252:         self.__updateDistanceLabel(str(round(distanceMoved, 2)) ,"encoder ticks")
253:         self.__update_position_label()
254:
255:
256:
257:
258:
259:     def __update(self):
260:         """
261:         updates tkinter canvas so robot can be seen moving
262:         and also allows for a pause
263:         """
264:         self.canvas.delete(self.square)
265:         self.canvas.delete(self.line)
266:
267:         self.square = self.canvas.create_polygon(self.squareVertexes,
268:                                                  outline="black",
269:                                                  fill="#949596",
270:                                                  width=3)
271:
272:         self.line = self.canvas.create_polygon(self.lineVertexes, width=3)
273:
274:
275:         if not self.controlPanelFrame.keepRunning: #allows for pause during turns
276:             while not self.controlPanelFrame.keepRunning:
277:                 time.sleep(0.1)
278:                 self.master.update()
279:
280:         self.master.update()
281:
282:
283:     def __updateDistanceLabel(self, distance=0, units=""):
284:         """
285:         updates distance travelled label
286:         """
287:         text = "distance moved: " + str(distance) + " " + units
288:         while len(text) < 38:
289:             text = text + " "
290:         self.robotInfoFrame.distanceMovedLabelText.set(text)
291:
292:         self.master.update()
293:
294:     def __update_orientation_label(self, units="degrees"):
295:         angle = self.orientationDegrees
296:         if units == "radians":
297:             angle = self.__to_radians(angle)
298:         text = "Orientation: " + str(angle)
299:         self.robotInfoFrame.orientationLabelText.set(text)
300:
301:
302:     def __update_position_label(self):
303:         x = round(self.inches(self.__calcCenters()[1]) - self.x_offset_in, 2)
304:         y = round(self.inches(self.__calcCenters()[0]) - self.y_offset_in, 2)
305:
306:         text = "(x: " + str(x) + ", y: " + str(y) + ")"
307:
308:         self.robotInfoFrame.positionLabelText.set(text)
309:
310:
311:     def __pixels(self, rotationUnits):
312:         """
313:         converts encoder ticks to pixels
314:         """
315:         revolutions = rotationUnits / 360
316:         inches = revolutions * (self.diameterOfWheel * math.pi)
317:         pixelsToMove = (inches * self.fieldSize) / 144
318:         # print(rotationUnits, pixelsToMove)
319:         return pixelsToMove
320:
321:
322:     def __encoderTicks(self, pixels):
323:         """
324:         converts pixels to encoder ticks
325:         """
326:         inches = pixels * (144 / self.fieldSize)
327:         revolutions = inches / (self.diameterOfWheel * math.pi)
328:         encoderTicks = revolutions * 360
329:
330:         return encoderTicks
331:
332:     def __to_radians(self, degrees):
333:         return ((degrees * math.pi) / 180)
334:
335:     def __to_degrees(self, radians):
336:         return ((radians * 180) / math.pi)
337:
338:     def __in_to_encoder_ticks(self, inches):
339:         circumference = (self.diameterOfWheel * math.pi);
```

```
340:         revolutions = inches / circumference;
341:         encoder_ticks = revolutions * 360;
342:
343:         return encoder_ticks;
344:
345:
346:     def inches(self, pixels):
347:         """
348:         converts pixels to inches
349:         """
350:         inches = pixels * (144 / self.fieldSize)
351:
352:         return inches
353:
354:
355:     def show(self, angle=0, position=[[0, 0], [0, 0]]):
356:         """
357:         starting function that shows the robot based on two coordinates
358:         if more than two coordinates are given use different show function
359:         """
360:
361:         #if len(position) > 2:
362:         if 1:
363:             x1 = position[0][0]
364:             y1 = position[0][1]
365:             x2 = position[1][0]
366:             y2 = position[1][1]
367:
368:             self.orientationDegrees = 90
369:             self.squareVertexes = [
370:                 [x1, y1],
371:                 [x2, y1],
372:                 [x2, y2],
373:                 [x1, y2]
374:                 ]
375:
376:             self.sizeOfSquare = abs(self.squareVertexes[0][0] - self.squareVertexes[1][0])
377:
378:             y2 = (y2 - y1) / 4
379:             self.lineVertexes = [
380:                 [x1, y1+y2],
381:                 [x2, y1+y2],
382:                 [x2, y1+y2+4],
383:                 [x1, y1+y2+4]
384:                 ]
385:
386: #       else:
387: #           x1 = position[0][0]
388: #           x2 = position[1][0]
389: #           x3 = position[2][0]
390: #           x4 = position[3][0]
391: #
392: #           y1 = position[0][1]
393: #           y2 = position[1][1]
394: #           y3 = position[2][1]
395: #           y4 = position[3][1]
396: #
397: #           self.squareVertexes = position
398: #           self.lineVertexes = []
399:
400:
401:         self.square = self.canvas.create_polygon(self.squareVertexes,
402:                                                  outline="black",
403:                                                  fill="#949596",
404:                                                  width=3)
405:
406:         turn = 90 - (angle % 360)
407:
408:         self.line = self.canvas.create_polygon(self.lineVertexes, width=3)
409:         self.__rotateInPlace(turn)
410:         self.__update()
411:         self.__generate_postscript()
412:
413:         self.orientationDegrees = angle % 360
414:
415:
416:         self.x_offset_in = self.inches(self.__calcCenters()[1])
417:         self.y_offset_in = self.inches(self.__calcCenters()[0])
418:         self.theta_offset_deg = self.orientationDegrees
419:
420:
421:
422:
423:     def reverse(self):
424:         """
425:         reverses orientation of the robot
426:         """
427:         self.reversed = not(self.reversed)
428:
429:
430:
431:
432:     def forward(self, rotationUnits):
433:         """
434:         moves the robot forward and straight
435:         """
436:         self.robotInfoFrame.commandLabelText.set(("forward " + str(rotationUnits)))
437:
438:         if self.reversed:
439:             rotationUnits = 0 - rotationUnits
440:
441:         pixelsToMove = self.__pixels((rotationUnits))
442:
443:         self.__move(pixelsToMove)
444:         self.__update()
445:
446:
447:
448:
449:     def backward(self, rotationUnits):
450:         """
451:         moves the robot backwards and straight
452:         """
```

```
453:        self.robotInfoFrame.commandLabelText.set(("backward " + str(rotationUnits)))
454:
455:        rotationUnits = 0 - rotationUnits
456:        if self.reversed:
457:            rotationUnits = 0 - rotationUnits
458:
459:        pixelsToMove = self.__pixels(rotationUnits)
460:
461:
462:        self.__move(pixelsToMove)
463:        self.__update()
464:
465:
466:    def leftSide(self, angle):
467:        """
468:        turns the robot right so that only one side is moving
469:        """
470:        self.robotInfoFrame.commandLabelText.set(("leftSide " + str(angle)))
471:
472:        if self.reversed:
473:            angle = 0 - angle
474:
475:        pivotPoints = self.squareVertexes[2]
476:
477:        turned = 0
478:        orientation = angle / abs(angle) #to account for negative turns
479:        toMove = orientation * .5
480:        while turned < abs(angle): #turn to specified angle
481:            self.__rotate(toMove, pivotPoints)
482:            self.__update()
483:            self.__generate_postscript()
484:
485:            time.sleep(self.__calcSleepTime(angle, angle))
486:
487:            turned += .5
488:            self.__updateDistanceLabel(str(round(turned, 2)), "degrees")
489:
490:            self.orientationDegrees = (self.orientationDegrees - toMove) % 360
491:            self.robotInfoFrame.orientationLabelText.set("orientation: " + str(self.orientationDegrees))
492:            self.__update_position_label()
493:
494:
495:    def rightSide(self, angle):
496:        """
497:        turns the robot left so that only the right side is moving
498:        """
499:        self.robotInfoFrame.commandLabelText.set(("rightSide " + str(angle)))
500:
501:        angle = 0 - angle
502:        if self.reversed:
503:            angle = 0 - angle
504:
505:        pivotPoints = self.squareVertexes[3]
506:
507:        turned = 0
508:        orientation = angle / abs(angle) #to account for negative turns
509:        toMove = (-1 * orientation * .5)
510:        while turned < abs(angle): #turn to specified angle
511:            self.__rotate(toMove, pivotPoints)
512:            self.__update()
513:            self.__generate_postscript()
514:
515:            time.sleep(self.__calcSleepTime(angle, angle))
516:            turned += .5
517:            self.__updateDistanceLabel(str(round(turned, 2)), "degrees")
518:
519:            self.orientationDegrees = (self.orientationDegrees + toMove) % 360
520:            self.__update_orientation_label()
521:            self.__update_position_label()
522:
523:
524:
525:    def turnLeft(self, angle):
526:        """
527:        turn in place left
528:        """
529:        self.robotInfoFrame.commandLabelText.set(("turnLeft " + str(angle)))
530:
531:        if self.reversed:
532:            angle = 0 - angle
533:
534:        turned = 0
535:        orientation = angle / abs(angle) #to account for negative turns
536:        toMove = -1 * orientation * .5
537:
538:        while turned < abs(angle): #turn to specified angle
539:            self.__rotateInPlace(toMove)
540:            self.__update()
541:            self.__generate_postscript()
542:
543:            time.sleep(self.__calcSleepTime(angle, angle))
544:            turned += .5
545:            self.__updateDistanceLabel(str(round(turned, 2)), "degrees")
546:
547:            self.orientationDegrees = (self.orientationDegrees + toMove) % 360
548:            self.__update_orientation_label()
549:            self.__update_position_label()
550:
551:
552:
553:
554:    def turnRight(self, angle):
555:        """
556:        turn in place right
557:        """
558:        self.robotInfoFrame.commandLabelText.set(("turnRight " + str(angle)))
559:
560:        if self.reversed:
561:            angle = 0 - angle
562:
563:        turned = 0
564:        orientation = angle / abs(angle) #to account for negative turns
565:        toMove = .5 * orientation
```

```
566:
567:        while turned < abs(angle): #turn to specified angle
568:            self.__rotateInPlace(toMove)
569:            self.__update()
570:            self.__generate_postscript()
571:
572:            time.sleep(self.__calcSleepTime(angle*2, angle))
573:            turned += .5
574:            self.__updateDistanceLabel(str(round(turned, 2)), "degrees")
575:
576:
577:            self.orientationDegrees = (self.orientationDegrees + toMove) % 360
578:            self.__update_orientation_label()
579:            self.__update_position_label()
580:
581:
582:
583:
584:    def drive_to_point(self, x, y, explicit_direction=0):
585:        """
586:        drive to a point
587:        """
588:        self.robotInfoFrame.commandLabelText.set(("drive_to_point (" + str(x) + ", " + str(y) + ")"))
589:
590:
591:        current_x = self.inches(self.__calcCenters()[1]) - self.x_offset_in
592:        current_y = self.inches(self.__calcCenters()[0]) - self.y_offset_in
593:        current_angle = self.__to_radians(self.orientationDegrees - self.theta_offset_deg)
594:
595:        end_x = x
596:        end_y = y
597:
598:
599:        dx = end_x - current_x
600:        dy = end_y - current_y
601:        print("current coords: ", current_x, current_y, self.orientationDegrees - self.theta_offset_deg)
602:        print("dx,dy:", dx, dy)
603:        print("end coords:", end_x, end_y)
604:        # print(self.__to_degrees(math.atan2(dy, dx)))
605:        dtheta = (math.atan2(dy, dx))
606:        if dtheta < 0:
607:            dtheta += 2 * math.pi
608:
609:        if current_angle < 0:
610:            current_angle += 2 * math.pi
611:
612:        current_angle = -current_angle + (math.pi/2)
613:        to_turn_face_forwards = current_angle - dtheta
614:        to_turn_face_backwards = to_turn_face_forwards - math.pi
615:        # print(self.__to_degrees(current_angle), self.__to_degrees(dtheta))
616:        if to_turn_face_forwards > math.pi:
617:            to_turn_face_forwards = (-2 * math.pi) + to_turn_face_forwards
618:        elif to_turn_face_forwards < -math.pi:
619:            to_turn_face_forwards = (2 * math.pi) + to_turn_face_forwards
620:
621:        if to_turn_face_backwards > math.pi:
622:            to_turn_face_backwards = (-2 * math.pi) + to_turn_face_backwards
623:        elif to_turn_face_backwards < -math.pi:
624:            to_turn_face_backwards = (2 * math.pi) + to_turn_face_backwards
625:
626:        if explicit_direction == 1:
627:            to_turn = to_turn_face_forwards
628:            direction = 1
629:        elif explicit_direction == -1:
630:            to_turn = to_turn_face_backwards
631:            direction = -1
632:        elif abs(to_turn_face_forwards) < abs(to_turn_face_backwards):
633:            to_turn = to_turn_face_forwards
634:            direction = 1
635:        else:
636:            to_turn = to_turn_face_backwards
637:            direction = -1
638:
639:        to_drive_inches = math.sqrt((dx**2) + (dy**2))
640:        to_drive_enc = direction * self.__in_to_encoder_ticks(to_drive_inches)
641:
642:
643:        # perform turn command
644:        angle = self.__to_degrees(to_turn)
645:        print("to turn: ", angle)
646:        if angle != 0:
647:            if self.reversed:
648:                angle = 0 - angle
649:
650:            turned = 0
651:            orientation = angle / abs(angle) #to account for negative turns
652:            toMove = .5 * orientation
653:
654:            while turned < abs(angle): #turn to specified angle
655:                self.__rotateInPlace(toMove)
656:                self.__update()
657:                self.__generate_postscript()
658:
659:                time.sleep(self.__calcSleepTime(angle*2, angle))
660:                turned += .5
661:                self.__updateDistanceLabel(str(round(turned, 2)), "degrees")
662:
663:
664:                self.orientationDegrees = (self.orientationDegrees + toMove) % 360
665:                self.__update_orientation_label()
666:                self.__update_position_label()
667:
668:
669:
670:        # perform drive command
671:        if self.reversed:
672:            to_drive_enc = 0 - to_drive_enc
673:
674:        pixelsToMove = self.__pixels((to_drive_enc))
675:
676:        self.__move(pixelsToMove)
677:        self.__update()
```

```
1:  #!/usr/bin/env python3
2:  # -*- coding: utf-8 -*-
3:
4:  import tkinter as tk
5:
6:  class runningFrame:
7:      """
8:      makes frame for running label
9:      """
10:     def __init__(self, master):
11:         self.master = master
12:
13:         self.guiFrame = tk.Frame(self.master)
14:         self.guiFrame.grid(row=2, column=0, sticky='news')
15:
16:
17:         ##### labels #####
18:
19:         #running label
20:         self.runningLabelText = tk.StringVar()
21:         self.runningLabel = tk.Label(self.guiFrame, textvariable=self.runningLabelText, font=("Courier", 15))
22:
23:
24:
25:     def placeObjects(self):
26:         """
27:         places labels
28:         """
29:
30:         self.runningLabel.grid(sticky='w', row=0, column=0)
31:
32:         self.guiFrame.grid_columnconfigure(0, weight=1)
```

```
1:  #!/usr/bin/env python3
2:  # -*- coding: utf-8 -*-
3:
4:  import time
5:
6:  class stopwatch:
7:
8:      def __init__(self):
9:          self.beginning = None
10:
11:     def start(self):
12:         """
13:         starts stopwatch
14:         """
15:         self.beginning = time.time()
16:
17:     def stop(self):
18:         """
19:         stops stopwatch and returns
20:         time elapsed
21:         """
22:         end = time.time()
23:         timeElapsed = end - self.beginning
24:
25:         return timeElapsed
```

1: **Update** *creds.json for google docs spreadsheet that will be modified*
2:
3: in "Scout.py" change sku in class constructor to sku of tournament

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 24 22:08:27 2019

@author: aiden
"""

import requests
import json
import stopwatch
from usefultools import split
import multiprocessing as mp
import gspread
from oauth2client.client import SignedJwtAssertionCredentials
import time


class getData:
    def __init__(self):
        self.sku = ""

        self.sheet_name = ""

        self.matchesWonUrl = 'https://api.vexdb.io/v1/get_matches?round=2'
        self.skillsUrl = 'https://api.vexdb.io/v1/get_skills?'
        self.pointsUrl = 'https://api.vexdb.io/v1/get_rankings?sku=' + self.sku

        self.teams = []
        self.allTeams = []
        self.ranges ={}

        self.matchesWonData = {}
        self.matchesWonOverallData = {}
        self.combinedSkillsScoreData = {}
        self.driverSkillsScoreData = {}
        self.pointsData = {}
        self.rankingData = {}

        self.NUM_PROCESSES = 20
        self.TEAM_NUM_COL = 2

        self.sheet = None


    def __calcWin(self, team, data):
        """
        calculates if a match was won or not by finding what
        color a team was and the score of the match
        it is a win if the teams color score more points
        """
        colors = ['red1', 'red2', 'blue1', 'blue2']
        teamColor = ''
        for color in colors:
            if data.get(color) == team:
                c = color.split('1')[0]
                c = c.split('2')[0]
                teamColor = c
                break

        redScore = data.get('redscore')
        blueScore = data.get('bluescore')

        if redScore != blueScore:
            if redScore > blueScore:
                winner = 'red'
            else:
                winner = 'blue'

            if teamColor == 'red' and winner == 'red':
                return 1
            elif teamColor == 'blue' and winner == 'blue':
                return 1
            else:
                return 0

        elif redScore == blueScore and (redScore != 0 and blueScore != 0):
            return -1

        else:
            return None



    def __getMatchesWon(self, teams, q):
        """
        gets win/loss/tie data for a team
        """
        for team in teams:
            team = team.split('\n')[0]
            url = self.matchesWonUrl + '&team=' + team + '&season=Tower Takeover'

            data = requests.get(url)
            if data.status_code == 200 and team != '':

                winStruct = {'wins':0,
                             'losses':0,
                             'ties':0
                            }

                data = json.loads(data.content.decode('utf-8'))
                data = data.get('result')

                for entry in data:
#                    print(team, entry)
                    x = self.__calcWin(team, entry)
                    if x == 1:
                        winStruct['wins'] = winStruct.get('wins') + 1
                    elif x == 0:
                        winStruct['losses'] = winStruct.get('losses') + 1
                    elif x == -1:
                        winStruct['ties'] = winStruct.get('ties') + 1
```

```
114:                 else:
115:                     pass
116:             q.put({team: winStruct})
117:
118:
119:
120:     def __getDriverSkillsScore(self, teams, q):
121:         """
122:         gets driver skills score for a team
123:         """
124:         for team in teams:
125:             team = team.split('\n')[0]
126:             url = self.skillsUrl + 'team=' + team + '&season=Tower Takeover&type=0'
127:
128:             data = requests.get(url)
129:             if data.status_code == 200:
130:
131:                 data = json.loads(data.content.decode('utf-8'))
132:                 data = data.get('result')
133:                 highest = 0
134:
135:                 for item in data:
136:                     value = item.get('score')
137:                     if value > highest:
138:                         highest = value
139:
140:                 q.put({team: highest})
141:
142:
143:     def __getCombinedSkillsScore(self, teams, q):
144:         """
145:         gets the combined skills score for a team
146:         (driver and programming)
147:         """
148:         for team in teams:
149:             team = team.split('\n')[0]
150:             url2 = self.skillsUrl + 'team=' + team + '&season=Tower Takeover&type=2'
151:
152:             data = requests.get(url2)
153:             if data.status_code == 200:
154:
155:
156:                 data = json.loads(data.content.decode('utf-8'))
157:                 data = data.get('result')
158:                 highest = 0
159:
160:                 for item in data:
161:                     value = item.get('score')
162:                     if value > highest:
163:                         highest = value
164:                 q.put({team: highest})
165:
166:
167:     def __getPoints(self, teams, q):
168:         """
169:         gets win points/auton points/strength points/calculated
170:         contribution to win margin data for a team
171:         """
172:         for team in teams:
173:             team = team.split('\n')[0]
174:             url = self.pointsUrl + '&team=' + team + '&season=Tower Takeover&type=2'
175:
176:             data = requests.get(url)
177:             if data.status_code == 200:
178:                 try:
179:                     data = json.loads(data.content.decode('utf-8'))
180:                     data = data.get('result')[0]
181:                     pointsStruct = {
182:                         'wp':data.get('wp'),
183:                         'ap':data.get('ap'),
184:                         'sp':data.get('sp'),
185:                         'ccwm':data.get('ccwm')
186:                         }
187:                     q.put({team: pointsStruct})
188:
189:                 except IndexError:
190:                     q.put({team: 'N/A'})
191:
192:     def __getRanking(self, teams, q):
193:         """
194:         gets the ranking of a team at an event
195:         """
196:         for team in teams:
197:             team = team.split('\n')[0]
198:             url = self.pointsUrl + '&team=' + team + '&season=Tower Takeover&type=2'
199:             #print(url)
200:
201:             data = requests.get(url)
202:             if data.status_code == 200:
203:
204:                 data = json.loads(data.content.decode('utf-8'))
205:                 try:
206:                     data = data.get('result')[0]
207:                     q.put({team: data.get('rank')})
208:
209:                 except IndexError:
210:                     q.put({team: 'N/A'})
211:
212:
213:     def __parralellise(self, func, returnDict):
214:         """
215:         processes the data in parrallel so that if there is a lot of
216:         teams the operation can be performed quickly
217:         """
218:         queues = []
219:         processes = []
220:         for i in range(len(self.teams)):
221:             queues.append(mp.Queue())
222:             p = mp.Process(target=func, args=(self.teams[i], queues[i],))
223:             processes.append(p)
224:             p.daemon = True
225:             p.start()
226:         for process in processes:
```

```
227:            process.join()
228:
229:        for q in queues:
230:            while not q.empty():
231:                returnDict.update(q.get(timeout=.1))
232:
233:
234:
235:
236:
237:    def openSheet(self):
238:        """
239:        opens the sheet
240:        need to change workbook to the sheet that will be edited
241:        and the sheet name to the sheet that corresponds with the
242:        data that will be collected
243:        """
244:        json_key = json.load(open('/home/aiden/Documents/google_credentials/creds.json'))
245:        scope = ['https://spreadsheets.google.com/feeds',
246:        'https://www.googleapis.com/auth/drive']
247:
248:        credentials = SignedJwtAssertionCredentials(json_key['client_email'], json_key['private_key'].encode(), scope)
249:
250:        file = gspread.authorize(credentials)
251:        workbook = file.open("536C_Scouting")
252:        self.sheet = workbook.worksheet(self.sheet_name)
253:
254:
255:
256:
257:
258:    def getTeams(self):
259:        """
260:        gets the teams by reading the data in the sheet
261:        this looks for data to find by comparing the first
262:        row to a list of valid headers then removes the headers
263:        from the list of cells to be updated
264:        """
265:        valid_headers = [
266:                'Rank',
267:                'WP/AP/SP/CCWM',
268:                'W-L-T (today)',
269:                'W-L-T (overall season)',
270:                'driver skills',
271:                'combined skills score'
272:                ]
273:
274:        self.allTeams = self.sheet.col_values(1) #column that teams are stored in
275:        self.allTeams.pop(0) #remove header
276:        headers = self.sheet.row_values(1) #remove header
277:        row = 2
278:        column = 1
279:        for header in headers:
280:            if header in valid_headers:
281:                start = chr(ord('@')+column) + str(row)
282:                end = chr(ord('@')+column) + str((len(self.allTeams) + 1))
283:                string = start + ':' + end
284:                range_= self.sheet.range(string)
285:                self.ranges.update({header:range_})
286:            column += 1
287:
288:        #limits num processes to the number of teams
289:        if self.NUM_PROCESSES > len(self.allTeams):
290:            self.NUM_PROCESSES = len(self.allTeams)
291:
292:        self.teams = list(split.split(self.allTeams, self.NUM_PROCESSES))
293:
294:
295:
296:
297:
298:    def collect(self):
299:        """
300:        collects all the data if that data is a valid header
301:        """
302:        if 'W-L-T (overall season)' in self.ranges.keys():
303:            self.__parralellise(self.__getMatchesWon, self.matchesWonOverallData)
304:
305:        self.matchesWonUrl = self.matchesWonUrl + '&sku=' + self.sku
306:
307:        if 'W-L-T (today)' in self.ranges.keys():
308:            self.__parralellise(self.__getMatchesWon, self.matchesWonData)
309:
310:        if 'combined skills score' in self.ranges.keys():
311:            self.__parralellise(self.__getCombinedSkillsScore, self.combinedSkillsScoreData)
312:        if 'driver skills' in self.ranges.keys():
313:            self.__parralellise(self.__getDriverSkillsScore, self.driverSkillsScoreData)
314:        if 'WP/AP/SP/CCWM' in self.ranges.keys():
315:            self.__parralellise(self.__getPoints, self.pointsData)
316:        if 'Rank' in self.ranges.keys():
317:            self.__parralellise(self.__getRanking, self.rankingData)
318:
319:
320:
321:    def printData(self):
322:        """
323:        prints the data for each team to be used for debugging
324:        purposes
325:        """
326:        if self.matchesWonData:
327:            #print("w-l-t")
328:            for team in self.matchesWonData.keys():
329:                try:
330:                    wins = self.matchesWonData.get(team)
331:                    record = [wins.get('wins'), wins.get('losses'), wins.get('ties')]
332:                    formattedRecord = str(record[0]) + '-' + str(record[1]) + '-' + str(record[2])
333:                    print(formattedRecord)
334:                except AttributeError:
335:                    print('N/A')
336:            for i in range(20):
337:                print("")
338:
339:        if self.combinedSkillsScoreData:
```

```
340:         print("combined skills")
341:         for team in self.combinedSkillsScoreData.keys():
342:             print(self.combinedSkillsScoreData.get(team))
343:
344:
345:         for i in range(20):
346:             print("")
347:
348:     if self.driverSkillsScoreData:
349:         print("driver skills")
350:         for team in self.driverSkillsScoreData.keys():
351:             print(self.driverSkillsScoreData.get(team))
352:
353:         for i in range(20):
354:             print("")
355:
356:
357:     if self.rankingData:
358:         print("ranking")
359:         for team in self.rankingData.keys():
360:             print(self.rankingData.get(team))
361:
362:         for i in range(20):
363:             print("")
364:
365:
366:     if self.pointsData:
367:         print("point values")
368:         for team in self.pointsData.keys():
369:             try:
370:                 val = self.pointsData.get(team)
371:                 points = (str(val.get('wp'))  + ' / '
372:                         + str(val.get('ap')) + ' / '
373:                         + str(val.get('sp')) + ' / '
374:                         + str(val.get('ccwm'))
375:                         )
376:                 print(points)
377:             except AttributeError:
378:                 print('N/A')
379:         for i in range(6):
380:             print("")
381:
382:
383:
384:
385:
386: def writeData(self):
387:     """
388:     writes the data in one chunk because the google api
389:     only allows so many edits
390:     by making it only one edit once all the data is collected
391:     this constraing can be worked around
392:     """
393:     cell_list = self.ranges.get('Rank') #write rank
394:     row = 0
395:     for cell in cell_list:
396:         try:
397:             team = self.allTeams[row].split('\n')[0]
398:             data = self.rankingData.get(team)
399:         except:
400:             data = "N/A"
401:         cell.value = data
402:         row += 1
403:
404:     self.sheet.update_cells(cell_list)
405:
406:
407:     cell_list = self.ranges.get('WP/AP/SP/CCWM') #write points
408:     row = 0
409:     for cell in cell_list:
410:         try:
411:             team = self.allTeams[row].split('\n')[0]
412:             val = self.pointsData.get(team)
413:             data = (str(val.get('wp'))  + ' / '
414:                     + str(val.get('ap')) + ' / '
415:                     + str(val.get('sp')) + ' / '
416:                     + str(val.get('ccwm'))
417:                     )
418:             cell.value = data
419:         except AttributeError:
420:             cell.value = 'N/A'
421:         except IndexError:
422:             cell.value = 'N/A'
423:         row += 1
424:     self.sheet.update_cells(cell_list)
425:
426:
427:
428:     cell_list = self.ranges.get('W-L-T (today)') #write w-l-t
429:     row = 0
430:     for cell in cell_list:
431:         try:
432:             team = self.allTeams[row].split('\n')[0]
433:             wins = self.matchesWonData.get(team)
434:             record = [wins.get('wins'), wins.get('losses'), wins.get('ties')]
435:             data = str(record[0]) + '-' + str(record[1]) + '-' + str(record[2])
436:
437:             cell.value = data
438:         except AttributeError:
439:             cell.value = 'N/A'
440:         row += 1
441:
442:     self.sheet.update_cells(cell_list)
443:
444:
445:
446:     cell_list = self.ranges.get('W-L-T (overall season)') #write w-l-t for
447:     row = 0                                 #overall season
448:     for cell in cell_list:
449:         try:
450:             team = self.allTeams[row].split('\n')[0]
451:             wins = self.matchesWonOverallData.get(team)
452:             record = [wins.get('wins'), wins.get('losses'), wins.get('ties')]
```

```
453:                data = str(record[0]) + '-' + str(record[1]) + '-' + str(record[2])
454:
455:                cell.value = data
456:            except AttributeError:
457:                cell.value = 'N/A'
458:            row += 1
459:
460:        self.sheet.update_cells(cell_list)
461:
462:
463:
464:
465:        cell_list = self.ranges.get('driver skills')#write driver skills
466:        row = 0
467:        for cell in cell_list:
468:            team = self.allTeams[row].split('\n')[0]
469:            data = self.driverSkillsScoreData.get(team)
470:            cell.value = data
471:
472:            row += 1
473:
474:        self.sheet.update_cells(cell_list)
475:
476:
477:
478:        cell_list = self.ranges.get('combined skills score') #write combined skills
479:        row = 0
480:        for cell in cell_list:
481:            team = self.allTeams[row].split('\n')[0]
482:            data = self.combinedSkillsScoreData.get(team)
483:            cell.value = data
484:
485:            row += 1
486:
487:        self.sheet.update_cells(cell_list)
488:
489:
490:
491:
492:
493:
494:
495:
496:    while 1: #collect data every so often
497:        stp = stopwatch.stopwatch()
498:        stp.start()
499:        d = getData()
500:        d.openSheet()
501:
502:        d.getTeams()
503:        d.collect()
504:        #print(d.rankingData)
505:        d.writeData()
506:        print("time taken:", stp.stop(), "sec")
507:
508:        time.sleep(45)
509:
510:
511:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Apr 28 11:36:52 2019

@author: aiden
"""
import requests
import json
import stopwatch
from usefultools import split
import multiprocessing as mp
import gspread
from oauth2client.client import SignedJwtAssertionCredentials
import time


class getTeams:
    """
    class for getting teams at a tournament
    based on the tournaments sku
    """
    def __init__(self):
        self.sheet_name = ""

        sku = ""

        self.url = "https://api.vexdb.io/v1/get_teams?round=5?&sku=" + sku
        self.elimsUrl = "https://api.vexdb.io/v1/get_teams?round=5?&sku=" + sku + '&matchnum='

        #legacy version that only works if matches have started
        #self.url = 'https://api.vexdb.io/v1/get_matches?round=2?&sku=' + sku
        #self.elimsUrl = 'https://api.vexdb.io/v1/get_matches?&sku=' + sku + '&matchnum='

        self.allTeams = []
        self.elimTeams = []

        self.COLLECT_ELIMS = 0
        self.COLLECT_TEAMS = 1

        self.NUM_PROCESSES = 50

        self.sheet = None

    def __getAllTeams(self, dicts, queue):
        """
        gets all teams that are registered
        """
        for entry in dicts:
            print(entry.get("number"))
            queue.put(entry.get("number"))



    def __getElimTeams(self, dicts, queue):
        """
        gets teams that are in the elimination matches
        does not work need to update
        """
        print(dicts)
        for entry in dicts:
            print(entry.get("number"))
            queue.put(entry.get("number"))


    def __parralellise(self, func, returnList, dicts):
        """
        starts threads that look through lists of entries
        for teams that are at the tournament
        this is used so that at events like worlds it does
        not take forever to run
        """
        queues = []
        processes = []
        for i in range(len(dicts)):
            queues.append(mp.Queue())
            p = mp.Process(target=func, args=(dicts[i], queues[i],))
            processes.append(p)
            p.daemon = True
            p.start()
        for process in processes:
            process.join()

        for q in queues:
            while not q.empty():
                returnList.append(q.get(timeout=.1))


    def collect(self):
        """
        collects the data from vexdb and splits it into
        entries based on self.NUM_PROCESSES so that data
        can be parsed faster especially for events like worlds
        """
        data = requests.get(self.url)
        if data.status_code == 200 and self.COLLECT_TEAMS:
            data = json.loads(data.content.decode('utf-8'))
            data = data.get('result')
            data = list(split.split(data, self.NUM_PROCESSES))

            self.__parralellise(self.__getAllTeams, self.allTeams, data)

            self.allTeams = list(set(self.allTeams))


        if self.COLLECT_ELIMS:
            links = []
            for num in range(1, 9):
                match = 'R16 #' + str(num) + '-1'
                link = self.elimsUrl + match
                links.append(link)

            data = []
```

```
114:
115:            for url in links:
116:                response = requests.get(url)
117:                if response.status_code == 200:
118:                    response = json.loads(response.content.decode('utf-8'))
119:                    allData = response.get('result') + data #merge lists
120:
121:
122:    #       data = requests.get(self.elimsUrl)
123:    #       if data.status_code == 200 and self.COLLECT_ELIMS:
124:    #           data = json.loads(data.content.decode('utf-8'))
125:    #           data = data.get('result')
126:            data = list(split.split(allData, self.NUM_PROCESSES))
127:
128:            self.__parralellise(self.__getElimTeams, self.elimTeams, data)
129:
130:            self.elimTeams = list(set(self.elimTeams))
131:
132:
133:
134:
135:
136:
137:
138:
139:    def printTeams(self):
140:        """
141:        prints the teams out
142:        used for debugging
143:        """
144:        for team in self.allTeams:
145:            print(team)
146:
147:        print("")
148:        print("")
149:
150:        for team in self.elimTeams:
151:            print(team)
152:
153:
154:
155:    def openSheet(self):
156:        """
157:        opens the sheet and loads the work book
158:        need to change the workbook to the file name
159:        and the sheet to the sheet that will be edited
160:        """
161:        json_key = json.load(open('/home/aiden/Documents/google_credentials/creds.json'))
162:        scope = scope = ['https://spreadsheets.google.com/feeds',
163:                         'https://www.googleapis.com/auth/drive']
164:
165:        credentials = SignedJwtAssertionCredentials(json_key['client_email'], json_key['private_key'].encode(), scope)
166:
167:        file = gspread.authorize(credentials)
168:        workbook = file.open("536C_Scouting")
169:        self.sheet = workbook.worksheet(self.sheet_name)
170:
171:
172:    def writeData(self, column):
173:        """
174:        writes the data in one chunk because the api only allows
175:        so many operations
176:        """
177:        if self.COLLECT_TEAMS:
178:            #leave room for header by setting to two and adding 1
179:            start = chr(ord('@')+column) + '2'
180:            end = chr(ord('@')+column) + str(len(self.allTeams) + 1)
181:            rang = start + ':' + end
182:            print(rang)
183:            cell_list = self.sheet.range(rang)
184:
185:            x = 0
186:            for cell in cell_list:
187:                cell.value = self.allTeams[x]
188:                x += 1
189:
190:            self.sheet.update_cells(cell_list)
191:
192:
193:        if self.COLLECT_ELIMS:
194:            start = chr(ord('@')+column) + '2'
195:            end = chr(ord('@')+column) + str(len(self.elimTeams) + 1)
196:            rang = start + ':' + end
197:            print(rang)
198:            cell_list = self.sheet.range(rang)
199:
200:            x = 0
201:            for cell in cell_list:
202:                cell.value = self.elimTeams[x]
203:                x += 1
204:
205:            self.sheet.update_cells(cell_list)
206:
207:
208:
209:
210:    g = getTeams()
211:    g.collect()
212:    g.printTeams()
213:    g.openSheet()
214:    print("sheet opened")
215:    g.writeData(1)
```

```
1:  #!/usr/bin/env python3
2:  # -*- coding: utf-8 -*-
3:
4:  import time
5:
6:  class stopwatch:
7:
8:      def __init__(self):
9:          self.beginning = None
10:
11:     def start(self):
12:         """
13:         starts stopwatch
14:         """
15:         self.beginning = time.time()
16:
17:     def stop(self):
18:         """
19:         stops stopwatch and returns
20:         time elapsed
21:         """
22:         end = time.time()
23:         timeElapsed = end - self.beginning
24:
25:         return timeElapsed
```

```
1: #!/bin/bash
2: prosv5 make
3: prosv5 upload --slot 2
4: prosv5 v5 run 2
5: prosv5 terminal
```

```
{
  "internal_motor_pid":[1, 0, 0, 0],
  "tilter_pid_consts":[1, 0, 0, 0],

  "front_right_port":20,
  "back_left_port":4,
  "front_left_port":18,
  "back_right_port":2,
  "left_intake_port":6,
  "right_intake_port":8,
  "tilter_port":19,
  "lift_port":9,

  "front_right_reversed":0,
  "back_left_reversed":1,
  "front_left_reversed":1,
  "back_right_reversed":0,
  "left_intake_reversed":1,
  "right_intake_reversed":0,
  "tilter_reversed":1,
  "lift_reversed":0,

  "tilter_setpoints":[100, 300, 400, 500],
  "lift_setpoints":[100, 300, 400, 500],
  "intake_speeds":[-63, -30, 0, 30, 63],

  "autons":{
    "auton1":"location_of_auton1.json",
    "auton2":"location_of_auton2.json",
    "auton3":"location_of_auton3.json",
    "auton4":"location_of_auton4.json"
  }
}
```

```
1: #!/bin/bash
2: echo "paste stack locations"
3: while true;
4: do
5:     read -p "" stack
6:     arm-none-eabi-addr2line --demangle --inlines -faps -e bin/monolith.elf $stack
7: done
8:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Mar  1 13:05:32 2020

@author: aiden
"""

import datetime
import glob

header_files = glob.glob("src/**/" + '*.hpp', recursive=True)
impl_files = glob.glob("src/**/" + '*.cpp', recursive=True)

files = sorted(header_files + impl_files)
todo_comments = {}
review_dates = {}
for file in files:
    todo_comments.update({file:[]})
    with open(file) as f:
        for line in f.readlines():
            if "TODO: " in line:
                comment = line.split("TODO: ")[-1].strip()
                todo_comments[file].append(comment)
            elif "@reviewed_on: " in line:
                date = line.split("@reviewed_on: ")[-1].strip()
                if date:
                    try:
                        date = datetime.datetime.strptime(date, "%m/%d/%Y")
                    except ValueError:
                        date = datetime.datetime.strptime(date, "%m/%d/%y")
                else:
                    date = False
                review_dates.update({file:date})

i1 = 0
i2 = 0

for file in todo_comments:
    if todo_comments.get(file):
        print(file)
        for comment in todo_comments.get(file):
            print("\t" + comment)
            i1 += 1

to_review = []
for file in review_dates:
    if not review_dates.get(file):  # skip if no date is provided
        to_review.append([file, 999999999])
        continue
    days_elapsed = abs(datetime.datetime.now() - review_dates.get(file)).days
    if days_elapsed > 30:
        to_review.append([file, days_elapsed])

print("\n")

to_review = sorted(to_review, key=lambda x: (x[1], x[0]))
for file in to_review:
    if file[1] == 999999999:
        print(file[0], ": ", "never", sep="")
    else:
        print(file[0], ": ", file[1], " days", sep="")
    i2 += 1

print("\n")

print("number of todo comments:", i1)
print("number of files needing review:", i2)

```

```
24953 [INFO] 24953 Byte read from stdin: ‘
24953 [INFO] 24953 Byte read from stdin: s
24953 [INFO] 24953 Byte read from stdin: o
24953 [INFO] 24953 Byte read from stdin: u
24953 [INFO] 24953 Byte read from stdin: t
24953 [INFO] 24953 Byte read from stdin: -
24953 [INFO] 24953 Byte read from stdin: 3
24953 [INFO] 24953 Byte read from stdin: .
24953 [INFO] 24953 Byte read from stdin: 8
24953 [INFO] 24953 Byte read from stdin: 6
24953 [INFO] 24953 Byte read from stdin: 4
24953 [INFO] 24953 Byte read from stdin: 5
24953 [INFO] 24953 Byte read from stdin: 3
24953 [INFO] 24953 Byte read from stdin:
24953 [INFO] 24953 Byte read from stdin: 3
24953 [INFO] 24953 Byte read from stdin: .


 _____ _____  ____ _____
| __ \| __ \ / __ \ / ____|        Powered by PROS for VEX V5
| |__) | |__) | |  | | (___     Version:                3.3.1
|  ___/|  _  /| |  | |\___ \    Uptime:                 24.974 s
| |    | | \ \| |__| |____) |   Compiled:               Unknown
|_|    |_|  \_\\____/|_____/    Directory:              Unknown


27766 [INFO], 2, motor added at 0x38e63d8
27766 [INFO], 2, motor added at 0x38e6370
27766 [INFO], 2, motor added at 0x38e6308
27766 [INFO], 2, motor added at 0x38e66a8
27766 [INFO], 2, motor added at 0x38e6440
27766 [INFO], 2, motor added at 0x38e64a8
27766 [INFO], 2, motor added at 0x38e6578
27766 [INFO], 2, motor added at 0x38e6510
27766 [INFO] CHASSIS_PID_TURN, Time: 2118, Actual_Vol1: 6.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 0.000000, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.000000, Absolute Angle: -0.000144, error history: 1, history size: 20, time out time: -2147481531, error difference: 0.000000, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27766 [INFO] CHASSIS_PID_TURN, Time: 2128, Actual_Vol1: -6.000000, Actual_Vol2: -6.000000, Actual_Vol3: -12.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 900.011435, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.001143, Absolute Angle: -0.000164, error history: 2, history size: 20, time out time: -2147481531, error difference: 0.001143, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27766 [INFO] CHASSIS_PID_TURN, Time: 2138, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1800.035220, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.002378, Absolute Angle: -0.000186, error history: 3, history size: 20, time out time: -2147481531, error difference: 0.002378, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27768 [INFO] CHASSIS_PID_TURN, Time: 2148, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: -12.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2700.071856, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.003664, Absolute Angle: -0.000208, error history: 4, history size: 20, time out time: -2147481531, error difference: 0.003664, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27768 [INFO] CHASSIS_PID_TURN, Time: 2158, Actual_Vol1: 0.000000, Actual_Vol2: 6.000000, Actual_Vol3: 74.000000, Actual_Vol4: -80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3600.122100, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.005024, Absolute Angle: -0.000232, error history: 5, history size: 20, time out time: -2147481531, error difference: 0.005024, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27768 [INFO] CHASSIS_PID_TURN, Time: 2168, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 277.000000, Actual_Vol4: -277.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4500.187036, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.006494, Absolute Angle: -0.000258, error history: 6, history size: 20, time out time: -2147481531, error difference: 0.006494, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27770 [INFO] CHASSIS_PID_TURN, Time: 2178, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 320.000000, Actual_Vol4: -320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5400.267413, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.008038, Absolute Angle: -0.000285, error history: 7, history size: 20, time out time: -2147481531, error difference: 0.008038, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27770 [INFO] CHASSIS_PID_TURN, Time: 2188, Actual_Vol1: 265.000000, Actual_Vol2: -259.000000, Actual_Vol3: 653.000000, Actual_Vol4: -659.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6300.364161, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.009675, Absolute Angle: -0.000313, error history: 8, history size: 20, time out time: -2147481531, error difference: 0.009675, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 8.200000, Actual_Vel4: 0.000000
27770 [INFO] CHASSIS_PID_TURN, Time: 2198, Actual_Vol1: 499.000000, Actual_Vol2: -499.000000, Actual_Vol3: 770.000000, Actual_Vol4: -912.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7200.477785, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.011362, Absolute Angle: -0.000343, error history: 9, history size: 20, time out time: -2147481531, error difference: 0.011362, over slew: 1, Actual_Vel1: 20.800000, Actual_Vel2: -0.000000, Actual_Vel3: 12.200000, Actual_Vel4: -12.200000
27772 [INFO] CHASSIS_PID_TURN, Time: 2208, Actual_Vol1: 795.000000, Actual_Vol2: -869.000000, Actual_Vol3: 1195.000000, Actual_Vol4: -1220.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8100.608902, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.013112, Absolute Angle: -0.000373, error history: 10, history size: 20, time out time: -2147481531, error difference: 0.013112, over slew: 1, Actual_Vel1: 20.800000, Actual_Vel2: -0.000000, Actual_Vel3: 17.400000, Actual_Vel4: -18.400000
27772 [INFO] CHASSIS_PID_TURN, Time: 2218, Actual_Vol1: 1318.000000, Actual_Vol2: -1392.000000, Actual_Vol3: 1540.000000, Actual_Vol4: -1614.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9000.758792, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.014989, Absolute Angle: -0.000406, error history: 11, history size: 20, time out time: -2147481531, error difference: 0.014989, over slew: 1, Actual_Vel1: 20.000000, Actual_Vel2: -7.200000, Actual_Vel3: 36.200000, Actual_Vel4: -31.400000
27772 [INFO] CHASSIS_PID_TURN, Time: 2228, Actual_Vol1: 1836.000000, Actual_Vol2: -1829.000000, Actual_Vol3: 1940.000000, Actual_Vol4: -2088.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9900.927480, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.016869, Absolute Angle: -0.000439, error history: 12, history size: 20, time out time: -2147481531, error difference: 0.016869, over slew: 1, Actual_Vel1: 55.200000, Actual_Vel2: -19.800000, Actual_Vel3: 70.000000, Actual_Vel4: -56.000000
27774 [INFO] CHASSIS_PID_TURN, Time: 2238, Actual_Vol1: 2285.000000, Actual_Vol2: -2353.000000, Actual_Vol3: 2248.000000, Actual_Vol4: -2384.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10801.111978, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.018450, Absolute Angle: -0.000466, error history: 13, history size: 20, time out time: -2147481531, error difference: 0.018450, over slew: 1, Actual_Vel1: 91.200000, Actual_Vel2: -51.600000, Actual_Vel3: 88.000000, Actual_Vel4: -90.200000
27774 [INFO] CHASSIS_PID_TURN, Time: 2248, Actual_Vol1: 2643.000000, Actual_Vol2: -2920.000000, Actual_Vol3: 2513.000000, Actual_Vol4: -2766.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11701.312031, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.020005, Absolute Angle: -0.000493, error history: 14, history size: 20, time out time: -2147481531, error difference: 0.020005, over slew: 1, Actual_Vel1: 105.800000, Actual_Vel2: -81.400000, Actual_Vel3: 101.000000, Actual_Vel4: -106.800000
27774 [INFO] CHASSIS_PID_TURN, Time: 2258, Actual_Vol1: 2914.000000, Actual_Vol2: -3333.000000, Actual_Vol3: 2834.000000, Actual_Vol4: -3006.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12601.525177, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.021315, Absolute Angle: -0.000516, error history: 15, history size: 20, time out time: -2147481531, error difference: 0.021315, over slew: 1, Actual_Vel1: 131.800000, Actual_Vel2: -105.400000, Actual_Vel3: 74.200000, Actual_Vel4: -131.600000
27776 [INFO] CHASSIS_PID_TURN, Time: 2268, Actual_Vol1: 3049.000000, Actual_Vol2: -3653.000000, Actual_Vol3: 3616.000000, Actual_Vol4: -3160.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13501.632044, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.010687, Absolute Angle: -0.000331, error history: 16, history size: 20, time out time: -2147481531, error difference: 0.021315, over slew: 1, Actual_Vel1: 126.800000, Actual_Vel2: -96.800000, Actual_Vel3: 35.600000, Actual_Vel4: -117.400000
27776 [INFO] CHASSIS_PID_TURN, Time: 2278, Actual_Vol1: 3129.000000, Actual_Vol2: -3881.000000, Actual_Vol3: 4306.000000, Actual_Vol4: -3345.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14401.451417, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.018063, Absolute Angle: 0.000171, error history: 17, history size: 20, time out time: -2147481531, error difference: 0.039377, over slew: 1, Actual_Vel1: 103.600000, Actual_Vel2: -87.200000, Actual_Vel3: 25.600000, Actual_Vel4: -83.600000
27776 [INFO] CHASSIS_PID_TURN, Time: 2288, Actual_Vol1: 3511.000000, Actual_Vol2: -4503.000000, Actual_Vol3: 4990.000000, Actual_Vol4: -3665.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15300.670815, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.078060, Absolute Angle: 0.001218, error history: 18, history size: 20, time out time: -2147481531, error difference: 0.099375, over slew: 1, Actual_Vel1: 68.800000, Actual_Vel2: -33.600000, Actual_Vel3: 33.400000, Actual_Vel4: -70.800000
27778 [INFO] CHASSIS_PID_TURN, Time: 2298, Actual_Vol1: 4121.000000, Actual_Vol2: -4977.000000, Actual_Vol3: 5624.000000, Actual_Vol4: -4059.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16198.571932, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.209888, Absolute Angle: 0.003519, error history: 19, history size: 20, time out time: -2147481531, error difference: 0.231203, over slew: 1, Actual_Vel1: 28.600000, Actual_Vel2: -33.600000, Actual_Vel3: 57.200000, Actual_Vel4: -79.600000
27778 [INFO] CHASSIS_PID_TURN, Time: 2308, Actual_Vol1: 4916.000000, Actual_Vol2: -5735.000000, Actual_Vol3: 6222.000000, Actual_Vol4: -4491.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17094.890460, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.368147, Absolute Angle: 0.006281, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.389462, over slew: 1, Actual_Vel1: 65.000000, Actual_Vel2: -23.400000, Actual_Vel3: 64.200000, Actual_Vel4: -90.200000
27778 [INFO] CHASSIS_PID_TURN, Time: 2318, Actual_Vol1: 5433.000000, Actual_Vol2: -6462.000000, Actual_Vol3: 6733.000000, Actual_Vol4: -4792.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17989.316281, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.557418, Absolute Angle: 0.009585, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.578732, over slew: 1, Actual_Vel1: 79.800000, Actual_Vel2: -52.400000, Actual_Vel3: 78.400000, Actual_Vel4: -111.000000
27780 [INFO] CHASSIS_PID_TURN, Time: 2328, Actual_Vol1: 5883.000000, Actual_Vol2: -6991.000000, Actual_Vol3: 7164.000000, Actual_Vol4: -5094.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18881.531910, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 1.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.778437, Absolute Angle: 0.013442, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.799752, over slew: 1, Actual_Vel1: 69.200000, Actual_Vel2: -57.200000, Actual_Vel3: 101.600000, Actual_Vel4: -138.600000
27780 [INFO] CHASSIS_PID_TURN, Time: 2338, Actual_Vol1: 6437.000000, Actual_Vol2: -7626.000000, Actual_Vol3: 7515.000000, Actual_Vol4: -5291.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19769.960788, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 2.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 1.157112, Absolute Angle: 0.020051, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.178427, over slew: 1, Actual_Vel1: 72.200000, Actual_Vel2: -56.800000, Actual_Vel3: 111.600000, Actual_Vel4: -162.200000
27780 [INFO] CHASSIS_PID_TURN, Time: 2348, Actual_Vol1: 6942.000000, Actual_Vol2: -8020.000000, Actual_Vol3: 7872.000000, Actual_Vol4: -5390.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20653.967137, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 4.000000, position_r: -1.000000, Heading_Sp: 90.000000, Relative_Heading: 1.599365, Absolute Angle: 0.027770, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.620680, over slew: 1, Actual_Vel1: 100.600000, Actual_Vel2: -62.200000, Actual_Vel3: 120.800000, Actual_Vel4: -187.400000
27782 [INFO] CHASSIS_PID_TURN, Time: 2358, Actual_Vol1: 7330.000000, Actual_Vol2: -6376.000000, Actual_Vol3: 7928.000000, Actual_Vol4: -5371.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21531.034134, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 5.000000, position_r: -2.000000, Heading_Sp: 90.000000, Relative_Heading: 2.293300, Absolute Angle: 0.039881, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.314615, over slew: 1, Actual_Vel1: 114.000000, Actual_Vel2: -63.600000, Actual_Vel3: 144.400000, Actual_Vel4: -193.400000
27782 [INFO] CHASSIS_PID_TURN, Time: 2368, Actual_Vol1: 7669.000000, Actual_Vol2: -6252.000000, Actual_Vol3: 7706.000000, Actual_Vol4: -5125.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22401.649649, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 8.000000, position_r: -2.000000, Heading_Sp: 90.000000, Relative_Heading: 2.938449, Absolute Angle: 0.051141, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.959763, over slew: 1, Actual_Vel1: 135.000000, Actual_Vel2: -46.400000, Actual_Vel3: 146.200000, Actual_Vel4: -180.200000
```

62: 27782 [INFO] CHASSIS_PID_TURN, Time: 2378, Actual_Vol1: 7774.000000, Actual_Vol2: -6733.000000, Actual_Vol3: 7632.000000, Actual_Vol4: -4971.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23264.271918, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 10.000000, position_r: -2.000000, Heading_Sp: 90.000000, Relative_Heading: 3.737773, Absolute Angle: 0.065092, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.759088, over slew: 1, Actual_Vel1: 145.800000, Actual_Vel2: -58.000000, Actual_Vel3: 140.200000, Actual_Vel4: -173.200000

63: 27784 [INFO] CHASSIS_PID_TURN, Time: 2388, Actual_Vol1: 7903.000000, Actual_Vol2: -6930.000000, Actual_Vol3: 7798.000000, Actual_Vol4: -4601.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24119.287181, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 13.000000, position_r: -3.000000, Heading_Sp: 90.000000, Relative_Heading: 4.498474, Absolute Angle: 0.078369, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.519788, over slew: 1, Actual_Vel1: 168.200000, Actual_Vel2: -84.200000, Actual_Vel3: 159.400000, Actual_Vel4: -144.200000

64: 27784 [INFO] CHASSIS_PID_TURN, Time: 2398, Actual_Vol1: 7835.000000, Actual_Vol2: -7503.000000, Actual_Vol3: 7798.000000, Actual_Vol4: -4571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24964.056953, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 16.000000, position_r: -4.000000, Heading_Sp: 90.000000, Relative_Heading: 5.523023, Absolute Angle: 0.096251, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.544337, over slew: 1, Actual_Vel1: 168.800000, Actual_Vel2: -94.800000, Actual_Vel3: 154.200000, Actual_Vel4: -140.000000

65: 27784 [INFO] CHASSIS_PID_TURN, Time: 2408, Actual_Vol1: 7983.000000, Actual_Vol2: -7232.000000, Actual_Vol3: 8039.000000, Actual_Vol4: -4564.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25797.979380, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 19.000000, position_r: -5.000000, Heading_Sp: 90.000000, Relative_Heading: 6.607757, Absolute Angle: 0.115183, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.629072, over slew: 1, Actual_Vel1: 183.800000, Actual_Vel2: -99.600000, Actual_Vel3: 180.200000, Actual_Vel4: -139.200000

66: 27786 [INFO] CHASSIS_PID_TURN, Time: 2418, Actual_Vol1: 7694.000000, Actual_Vol2: -7515.000000, Actual_Vol3: 8193.000000, Actual_Vol4: -4817.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26620.547837, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 22.000000, position_r: -6.000000, Heading_Sp: 90.000000, Relative_Heading: 7.743154, Absolute Angle: 0.134999, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.764469, over slew: 1, Actual_Vel1: 183.600000, Actual_Vel2: -97.800000, Actual_Vel3: 201.400000, Actual_Vel4: -155.600000

67: 27786 [INFO] CHASSIS_PID_TURN, Time: 2428, Actual_Vol1: 7503.000000, Actual_Vol2: -7373.000000, Actual_Vol3: 8082.000000, Actual_Vol4: -4854.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27431.227228, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 25.000000, position_r: -6.000000, Heading_Sp: 90.000000, Relative_Heading: 8.932061, Absolute Angle: 0.155750, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.953376, over slew: 1, Actual_Vel1: 187.000000, Actual_Vel2: -110.600000, Actual_Vel3: 203.400000, Actual_Vel4: -168.800000

68: 27786 [INFO] CHASSIS_PID_TURN, Time: 2438, Actual_Vol1: 7226.000000, Actual_Vol2: -7614.000000, Actual_Vol3: 7706.000000, Actual_Vol4: -4805.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28230.461524, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 29.000000, position_r: -8.000000, Heading_Sp: 90.000000, Relative_Heading: 10.076570, Absolute Angle: 0.175725, error history: 20, history size: 20, time out time: -2147481531, error difference: 10.097885, over slew: 1, Actual_Vel1: 197.000000, Actual_Vel2: -124.200000, Actual_Vel3: 195.000000, Actual_Vel4: -174.000000

69: 27788 [INFO] CHASSIS_PID_TURN, Time: 2448, Actual_Vol1: 7059.000000, Actual_Vol2: -7595.000000, Actual_Vol3: 7343.000000, Actual_Vol4: -4583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29014.765327, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 33.000000, position_r: -8.000000, Heading_Sp: 90.000000, Relative_Heading: 11.569620, Absolute Angle: 0.201784, error history: 20, history size: 20, time out time: -2147481531, error difference: 11.590934, over slew: 1, Actual_Vel1: 192.400000, Actual_Vel2: -123.600000, Actual_Vel3: 188.000000, Actual_Vel4: -180.200000

70: 27788 [INFO] CHASSIS_PID_TURN, Time: 2458, Actual_Vol1: 6733.000000, Actual_Vol2: -7922.000000, Actual_Vol3: 7102.000000, Actual_Vol4: -4410.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29785.610472, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 37.000000, position_r: -9.000000, Heading_Sp: 90.000000, Relative_Heading: 12.915486, Absolute Angle: 0.225274, error history: 20, history size: 20, time out time: -2147481531, error difference: 12.926172, over slew: 1, Actual_Vel1: 188.600000, Actual_Vel2: -124.600000, Actual_Vel3: 199.400000, Actual_Vel4: -183.000000

71: 27788 [INFO] CHASSIS_PID_TURN, Time: 2468, Actual_Vol1: 6548.000000, Actual_Vol2: -7657.000000, Actual_Vol3: 6856.000000, Actual_Vol4: -4300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30541.645449, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 42.000000, position_r: -10.000000, Heading_Sp: 90.000000, Relative_Heading: 14.396502, Absolute Angle: 0.251122, error history: 20, history size: 20, time out time: -2147481531, error difference: 14.378440, over slew: 1, Actual_Vel1: 185.800000, Actual_Vel2: -116.200000, Actual_Vel3: 221.800000, Actual_Vel4: -190.400000

72: 27790 [INFO] CHASSIS_PID_TURN, Time: 2478, Actual_Vol1: 6271.000000, Actual_Vol2: -7614.000000, Actual_Vol3: 6462.000000, Actual_Vol4: -4090.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31281.596530, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 46.000000, position_r: -10.000000, Heading_Sp: 90.000000, Relative_Heading: 16.004892, Absolute Angle: 0.279194, error history: 20, history size: 20, time out time: -2147481531, error difference: 15.926832, over slew: 1, Actual_Vel1: 188.600000, Actual_Vel2: -125.000000, Actual_Vel3: 206.200000, Actual_Vel4: -192.600000

73: 27790 [INFO] CHASSIS_PID_TURN, Time: 2488, Actual_Vol1: 6043.000000, Actual_Vol2: -8002.000000, Actual_Vol3: 6098.000000, Actual_Vol4: -3838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32007.198633, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 50.000000, position_r: -12.000000, Heading_Sp: 90.000000, Relative_Heading: 17.439790, Absolute Angle: 0.304238, error history: 20, history size: 20, time out time: -2147481531, error difference: 17.229901, over slew: 1, Actual_Vel1: 206.800000, Actual_Vel2: -135.000000, Actual_Vel3: 190.200000, Actual_Vel4: -195.600000

74: 27790 [INFO] CHASSIS_PID_TURN, Time: 2498, Actual_Vol1: 5772.000000, Actual_Vol2: -8051.000000, Actual_Vol3: 5914.000000, Actual_Vol4: -3647.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32716.206349, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 54.000000, position_r: -13.000000, Heading_Sp: 90.000000, Relative_Heading: 19.099228, Absolute Angle: 0.333200, error history: 20, history size: 20, time out time: -2147481531, error difference: 18.731081, over slew: 1, Actual_Vel1: 210.000000, Actual_Vel2: -147.200000, Actual_Vel3: 194.000000, Actual_Vel4: -182.400000

75: 27792 [INFO] CHASSIS_PID_TURN, Time: 2508, Actual_Vol1: 5562.000000, Actual_Vol2: -7971.000000, Actual_Vol3: 5673.000000, Actual_Vol4: -3431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33409.465182, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 59.000000, position_r: -14.000000, Heading_Sp: 90.000000, Relative_Heading: 20.674117, Absolute Angle: 0.360687, error history: 20, history size: 20, time out time: -2147481531, error difference: 20.116699, over slew: 1, Actual_Vel1: 189.400000, Actual_Vel2: -136.000000, Actual_Vel3: 210.000000, Actual_Vel4: -179.600000

76: 27792 [INFO] CHASSIS_PID_TURN, Time: 2518, Actual_Vol1: 5217.000000, Actual_Vol2: -8242.000000, Actual_Vol3: 5365.000000, Actual_Vol4: -3259.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34085.853712, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 63.000000, position_r: -16.000000, Heading_Sp: 90.000000, Relative_Heading: 22.361147, Absolute Angle: 0.390131, error history: 20, history size: 20, time out time: -2147481531, error difference: 21.582710, over slew: 1, Actual_Vel1: 179.200000, Actual_Vel2: -153.000000, Actual_Vel3: 216.600000, Actual_Vel4: -178.800000

77: 27792 [INFO] CHASSIS_PID_TURN, Time: 2528, Actual_Vol1: 5039.000000, Actual_Vol2: -8267.000000, Actual_Vol3: 5026.000000, Actual_Vol4: -3018.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34745.199622, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 67.000000, position_r: -17.000000, Heading_Sp: 90.000000, Relative_Heading: 24.065409, Absolute Angle: 0.419876, error history: 20, history size: 20, time out time: -2147481531, error difference: 22.908297, over slew: 1, Actual_Vel1: 183.200000, Actual_Vel2: -142.600000, Actual_Vel3: 201.600000, Actual_Vel4: -172.600000

78: 27794 [INFO] CHASSIS_PID_TURN, Time: 2538, Actual_Vol1: 4829.000000, Actual_Vol2: -8193.000000, Actual_Vol3: 4682.000000, Actual_Vol4: -2852.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35388.418998, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 71.000000, position_r: -18.000000, Heading_Sp: 90.000000, Relative_Heading: 25.678062, Absolute Angle: 0.448022, error history: 20, history size: 20, time out time: -2147481531, error difference: 24.078697, over slew: 1, Actual_Vel1: 189.600000, Actual_Vel2: -138.400000, Actual_Vel3: 184.600000, Actual_Vel4: -169.400000

79: 27794 [INFO] CHASSIS_PID_TURN, Time: 2548, Actual_Vol1: 4614.000000, Actual_Vol2: -8013.000000, Actual_Vol3: 4392.000000, Actual_Vol4: -2772.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36015.456502, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 76.000000, position_r: -20.000000, Heading_Sp: 90.000000, Relative_Heading: 27.296250, Absolute Angle: 0.476265, error history: 20, history size: 20, time out time: -2147481531, error difference: 25.002949, over slew: 1, Actual_Vel1: 195.600000, Actual_Vel2: -143.000000, Actual_Vel3: 177.400000, Actual_Vel4: -162.000000

80: 27794 [INFO] CHASSIS_PID_TURN, Time: 2558, Actual_Vol1: 4226.000000, Actual_Vol2: -8217.000000, Actual_Vol3: 4183.000000, Actual_Vol4: -2729.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36624.418762, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 79.000000, position_r: -22.000000, Heading_Sp: 90.000000, Relative_Heading: 29.103774, Absolute Angle: 0.507812, error history: 20, history size: 20, time out time: -2147481531, error difference: 26.165325, over slew: 0, Actual_Vel1: 175.400000, Actual_Vel2: -153.200000, Actual_Vel3: 182.000000, Actual_Vel4: -162.200000

81: 27796 [INFO] CHASSIS_PID_TURN, Time: 2568, Actual_Vol1: 4084.000000, Actual_Vol2: -8402.000000, Actual_Vol3: 4010.000000, Actual_Vol4: -2667.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37217.332056, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 83.000000, position_r: -24.000000, Heading_Sp: 90.000000, Relative_Heading: 30.708671, Absolute Angle: 0.535823, error history: 20, history size: 20, time out time: -2147481531, error difference: 26.970898, over slew: 0, Actual_Vel1: 169.200000, Actual_Vel2: -164.000000, Actual_Vel3: 186.000000, Actual_Vel4: -155.800000

82: 27796 [INFO] CHASSIS_PID_TURN, Time: 2578, Actual_Vol1: 3850.000000, Actual_Vol2: -8279.000000, Actual_Vol3: 3819.000000, Actual_Vol4: -2661.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37793.115962, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 87.000000, position_r: -26.000000, Heading_Sp: 90.000000, Relative_Heading: 32.421609, Absolute Angle: 0.565720, error history: 20, history size: 20, time out time: -2147481531, error difference: 27.923136, over slew: 0, Actual_Vel1: 171.400000, Actual_Vel2: -162.200000, Actual_Vel3: 182.400000, Actual_Vel4: -153.800000

83: 27796 [INFO] CHASSIS_PID_TURN, Time: 2588, Actual_Vol1: 3665.000000, Actual_Vol2: -8273.000000, Actual_Vol3: 3671.000000, Actual_Vol4: -2624.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38351.699728, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 89.000000, position_r: -28.000000, Heading_Sp: 90.000000, Relative_Heading: 34.141623, Absolute Angle: 0.595739, error history: 20, history size: 20, time out time: -2147481531, error difference: 28.618601, over slew: 0, Actual_Vel1: 170.600000, Actual_Vel2: -174.400000, Actual_Vel3: 167.400000, Actual_Vel4: -150.000000

84: 27798 [INFO] CHASSIS_PID_TURN, Time: 2598, Actual_Vol1: 3591.000000, Actual_Vol2: -8642.000000, Actual_Vol3: 3542.000000, Actual_Vol4: -2581.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38895.055902, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 92.000000, position_r: -32.000000, Heading_Sp: 90.000000, Relative_Heading: 35.664383, Absolute Angle: 0.622317, error history: 20, history size: 20, time out time: -2147481531, error difference: 29.056625, over slew: 0, Actual_Vel1: 173.600000, Actual_Vel2: -174.400000, Actual_Vel3: 163.000000, Actual_Vel4: -147.800000

85: 27798 [INFO] CHASSIS_PID_TURN, Time: 2608, Actual_Vol1: 3437.000000, Actual_Vol2: -8667.000000, Actual_Vol3: 3456.000000, Actual_Vol4: -2563.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39420.325254, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 96.000000, position_r: -34.000000, Heading_Sp: 90.000000, Relative_Heading: 37.473065, Absolute Angle: 0.653884, error history: 20, history size: 20, time out time: -2147481531, error difference: 29.729910, over slew: 0, Actual_Vel1: 158.600000, Actual_Vel2: -172.400000, Actual_Vel3: 157.400000, Actual_Vel4: -144.800000

86: 27798 [INFO] CHASSIS_PID_TURN, Time: 2618, Actual_Vol1: 3351.000000, Actual_Vol2: -8285.000000, Actual_Vol3: 3400.000000, Actual_Vol4: -2593.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39928.568954, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 99.000000, position_r: -37.000000, Heading_Sp: 90.000000, Relative_Heading: 39.175630, Absolute Angle: 0.683599, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.243569, over slew: 0, Actual_Vel1: 143.600000, Actual_Vel2: -167.200000, Actual_Vel3: 161.800000, Actual_Vel4: -145.200000

87: 27800 [INFO] CHASSIS_PID_TURN, Time: 2628, Actual_Vol1: 3302.000000, Actual_Vol2: -8341.000000, Actual_Vol3: 3308.000000, Actual_Vol4: -2630.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40419.826318, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 102.000000, position_r: -40.000000, Heading_Sp: 90.000000, Relative_Heading: 40.874264, Absolute Angle: 0.713246, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.797693, over slew: 0, Actual_Vel1: 140.800000, Actual_Vel2: -172.000000, Actual_Vel3: 158.600000, Actual_Vel4: -143.800000

88: 27800 [INFO] CHASSIS_PID_TURN, Time: 2638, Actual_Vol1: 3382.000000, Actual_Vol2: -8846.000000, Actual_Vol3: 3179.000000, Actual_Vol4: -2587.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40894.107657, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 104.000000, position_r: -44.000000, Heading_Sp: 90.000000, Relative_Heading: 42.571866, Absolute Angle: 0.742875, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.002246, over slew: 0, Actual_Vel1: 148.400000, Actual_Vel2: -197.800000, Actual_Vel3: 150.600000, Actual_Vel4: -145.600000

89: 27800 [INFO] CHASSIS_PID_TURN, Time: 2648, Actual_Vol1: 3376.000000, Actual_Vol2: -8778.000000, Actual_Vol3: 3043.000000, Actual_Vol4: -2519.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41351.548194, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 107.000000, position_r: -46.000000, Heading_Sp: 90.000000, Relative_Heading: 44.255946, Absolute Angle: 0.772268, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.340461, over slew: 0, Actual_Vel1: 151.200000, Actual_Vel2: -186.800000, Actual_Vel3: 142.200000, Actual_Vel4: -142.600000

90: 27802 [INFO] CHASSIS_PID_TURN, Time: 2658, Actual_Vol1: 3203.000000, Actual_Vol2: -8575.000000, Actual_Vol3: 3012.000000, Actual_Vol4: -2482.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41793.148995, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 109.000000, position_r: -50.000000, Heading_Sp: 90.000000, Relative_Heading: 45.839920, Absolute Angle: 0.799913, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.444318, over slew: 0, Actual_Vel1: 145.000000, Actual_Vel2: -167.400000, Actual_Vel3: 143.800000, Actual_Vel4: -144.800000

91: 27802 [INFO] CHASSIS_PID_TURN, Time: 2668, Actual_Vol1: 3000.000000, Actual_Vol2: -8753.000000, Actual_Vol3: 3006.000000, Actual_Vol4: -2458.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42217.863240, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 111.000000, position_r: -54.000000, Heading_Sp: 90.000000, Relative_Heading: 47.528576, Absolute Angle: 0.829386, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.523684, over slew: 0, Actual_Vel1: 132.600000, Actual_Vel2: -185.400000, Actual_Vel3: 143.200000, Actual_Vel4: -137.600000

92: 27802 [INFO] CHASSIS_PID_TURN, Time: 2678, Actual_Vol1: 2981.000000, Actual_Vol2: -8310.000000, Actual_Vol3: 2963.000000, Actual_Vol4: -2433.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42625.742256, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 113.000000, position_r: -58.000000, Heading_Sp: 90.000000, Relative_Heading: 49.212098, Absolute Angle: 0.858769, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.772309, over slew: 0, Actual_Vel1: 132.000000, Actual_Vel2: -179.200000, Actual_Vel3: 141.000000, Actual_Vel4: -139.400000

93: 27804 [INFO] CHASSIS_PID_TURN, Time: 2688, Actual_Vol1: 2926.000000, Actual_Vol2: -8630.000000, Actual_Vol3: 2932.000000, Actual_Vol4: -2291.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43016.881858, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 114.000000, position_r: -61.000000, Heading_Sp: 90.000000, Relative_Heading: 50.886040, Absolute Angle: 0.887985, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.786811, over slew: 0, Actual_Vel1: 126.000000, Actual_Vel2: -201.600000, Actual_Vel3: 134.200000, Actual_Vel4: -133.800000

94: 27804 [INFO] CHASSIS_PID_TURN, Time: 2698, Actual_Vol1: 2907.000000, Actual_Vol2: -8772.000000, Actual_Vol3: 2797.000000, Actual_Vol4: -2211.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43393.273086, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 116.000000, position_r: -65.000000, Heading_Sp: 90.000000, Relative_Heading: 52.360877, Absolute Angle: 0.913725, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.686761, over slew: 0, Actual_Vel1: 133.000000, Actual_Vel2: -208.800000, Actual_Vel3: 125.200000, Actual_Vel4: -129.600000

95: 27804 [INFO] CHASSIS_PID_TURN, Time: 2708, Actual_Vol1: 2846.000000, Actual_Vol2: -8612.000000, Actual_Vol3: 2643.000000, Actual_Vol4: -2174.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43753.086264, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 116.000000, position_r: -70.000000, Heading_Sp: 90.000000, Relative_Heading: 54.018682, Absolute Angle: 0.942660, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.657535, over slew: 0, Actual_Vel1: 124.200000, Actual_Vel2: -192.400000, Actual_Vel3: 115.400000, Actual_Vel4: -123.400000

96: 27806 [INFO] CHASSIS_PID_TURN, Time: 2718, Actual_Vol1: 2821.000000, Actual_Vol2: -8297.000000, Actual_Vol3: 2729.000000, Actual_Vol4: -2113.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44097.249899, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 118.000000, position_r: -77.000000, Heading_Sp: 90.000000, Relative_Heading: 55.583636, Absolute Angle: 0.969973, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.518227, over slew: 0, Actual_Vel1: 115.000000, Actual_Vel2: -189.400000, Actual_Vel3: 110.800000, Actual_Vel4: -125.400000

97: 27806 [INFO] CHASSIS_PID_TURN, Time: 2728, Actual_Vol1: 2735.000000, Actual_Vol2: -7885.000000, Actual_Vol3: 2618.000000, Actual_Vol4: -1996.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44421.920395, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 118.000000, position_r: -82.000000, Heading_Sp: 90.000000, Relative_Heading: 57.532950, Absolute Angle: 1.003995, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.854888, over slew: 0, Actual_Vel1: 104.800000, Actual_Vel2: -198.400000, Actual_Vel3: 113.200000, Actual_Vel4: -119.000000

98: 27806 [INFO] CHASSIS_PID_TURN, Time: 2738, Actual_Vol1: 2661.000000, Actual_Vol2: -7361.000000, Actual_Vol3: 2452.000000, Actual_Vol4: -1879.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44731.175417, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 118.000000, position_r: -87.000000, Heading_Sp: 90.000000, Relative_Heading: 59.074498, Absolute Angle: 1.030900, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.778248, over slew: 0, Actual_Vel1: 97.000000, Actual_Vel2: -201.400000, Actual_Vel3: 105.800000, Actual_Vel4: -111.000000

99: 27808 [INFO] CHASSIS_PID_TURN, Time: 2748, Actual_Vol1: 2741.000000, Actual_Vol2: -6850.000000, Actual_Vol3: 2396.000000, Actual_Vol4: -1786.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45025.089547, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 119.000000, position_r: -91.000000, Heading_Sp: 90.000000, Relative_Heading: 60.608587, Absolute Angle: 1.057675, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.504810

3, over slew: 0, Actual_Vel1: 99.200000, Actual_Vel2: -173.400000, Actual_Vel3: 104.600000, Actual_Vel4: -108.000000
100: 27808 [INFO] CHASSIS_PID_TURN, Time: 2758, Actual_Vol1: 2723.000000, Actual_Vol2: -6326.000000, Actual_Vol3: 2378.000000, Actual_Vol4: -1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45303.874626, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 120.000000, position_r: -95.000000, Heading_Sp: 90.000000, Relative_Heading: 62.121492, Absolute Angle: 1.084080, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.412821, over slew: 0, Actual_Vel1: 98.800000, Actual_Vel2: -173.000000, Actual_Vel3: 106.800000, Actual_Vel4: -100.600000
101: 27808 [INFO] CHASSIS_PID_TURN, Time: 2768, Actual_Vol1: 2575.000000, Actual_Vol2: -5852.000000, Actual_Vol3: 2279.000000, Actual_Vol4: -1608.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45567.733878, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 120.000000, position_r: -101.000000, Heading_Sp: 90.000000, Relative_Heading: 63.614075, Absolute Angle: 1.110131, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.192466, over slew: 0, Actual_Vel1: 95.200000, Actual_Vel2: -164.800000, Actual_Vel3: 108.400000, Actual_Vel4: -94.000000
102: 27810 [INFO] CHASSIS_PID_TURN, Time: 2778, Actual_Vol1: 2556.000000, Actual_Vol2: -5359.000000, Actual_Vol3: 2187.000000, Actual_Vol4: -1565.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45815.884714, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 120.000000, position_r: -106.000000, Heading_Sp: 90.000000, Relative_Heading: 65.184916, Absolute Angle: 1.137547, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.043293, over slew: 0, Actual_Vel1: 98.400000, Actual_Vel2: -165.600000, Actual_Vel3: 102.200000, Actual_Vel4: -92.000000
103: 27810 [INFO] CHASSIS_PID_TURN, Time: 2788, Actual_Vol1: 2470.000000, Actual_Vol2: -4885.000000, Actual_Vol3: 2045.000000, Actual_Vol4: -1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46049.611903, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 121.000000, position_r: -110.000000, Heading_Sp: 90.000000, Relative_Heading: 66.627281, Absolute Angle: 1.162721, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.962898, over slew: 0, Actual_Vel1: 97.200000, Actual_Vel2: -158.000000, Actual_Vel3: 91.800000, Actual_Vel4: -98.000000
104: 27810 [INFO] CHASSIS_PID_TURN, Time: 2798, Actual_Vol1: 2365.000000, Actual_Vol2: -4386.000000, Actual_Vol3: 1885.000000, Actual_Vol4: -1417.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46269.140671, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 121.000000, position_r: -114.000000, Heading_Sp: 90.000000, Relative_Heading: 68.047123, Absolute Angle: 1.187502, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.574058, over slew: 0, Actual_Vel1: 91.000000, Actual_Vel2: -158.600000, Actual_Vel3: 83.000000, Actual_Vel4: -85.800000
105: 27812 [INFO] CHASSIS_PID_TURN, Time: 2808, Actual_Vol1: 2187.000000, Actual_Vol2: -3905.000000, Actual_Vol3: 1780.000000, Actual_Vol4: -1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46475.692715, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -117.000000, Heading_Sp: 90.000000, Relative_Heading: 69.344796, Absolute Angle: 1.210151, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.169166, over slew: 0, Actual_Vel1: 82.800000, Actual_Vel2: -144.400000, Actual_Vel3: 81.200000, Actual_Vel4: -95.000000
106: 27812 [INFO] CHASSIS_PID_TURN, Time: 2818, Actual_Vol1: 2094.000000, Actual_Vol2: -3055.000000, Actual_Vol3: 1700.000000, Actual_Vol4: -1207.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46661.433960, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -121.000000, Heading_Sp: 90.000000, Relative_Heading: 71.425876, Absolute Angle: 1.246472, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.551612, over slew: 0, Actual_Vel1: 73.200000, Actual_Vel2: -117.800000, Actual_Vel3: 77.600000, Actual_Vel4: -91.000000
107: 27812 [INFO] CHASSIS_PID_TURN, Time: 2828, Actual_Vol1: 2057.000000, Actual_Vol2: -2686.000000, Actual_Vol3: 1663.000000, Actual_Vol4: -979.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46835.470646, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -125.000000, Heading_Sp: 90.000000, Relative_Heading: 72.596331, Absolute Angle: 1.266901, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.024465, over slew: 0, Actual_Vel1: 65.000000, Actual_Vel2: -107.800000, Actual_Vel3: 72.800000, Actual_Vel4: -82.000000
108: 27814 [INFO] CHASSIS_PID_TURN, Time: 2838, Actual_Vol1: 1990.000000, Actual_Vol2: -2384.000000, Actual_Vol3: 1651.000000, Actual_Vol4: -825.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46998.214738, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -129.000000, Heading_Sp: 90.000000, Relative_Heading: 73.725591, Absolute Angle: 1.286610, error history: 20, history size: 20, time out time: -2147481531, error difference: 29.469645, over slew: 0, Actual_Vel1: 63.200000, Actual_Vel2: -99.000000, Actual_Vel3: 67.800000, Actual_Vel4: -75.600000
109: 27814 [INFO] CHASSIS_PID_TURN, Time: 2848, Actual_Vol1: 1854.000000, Actual_Vol2: -2057.000000, Actual_Vol3: 1639.000000, Actual_Vol4: -764.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47150.095294, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -132.000000, Heading_Sp: 90.000000, Relative_Heading: 74.811944, Absolute Angle: 1.305571, error history: 20, history size: 20, time out time: -2147481531, error difference: 28.972024, over slew: 0, Actual_Vel1: 55.600000, Actual_Vel2: -87.600000, Actual_Vel3: 64.400000, Actual_Vel4: -70.800000
110: 27814 [INFO] CHASSIS_PID_TURN, Time: 2858, Actual_Vol1: 2008.000000, Actual_Vol2: -1817.000000, Actual_Vol3: 1466.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47292.502181, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -135.000000, Heading_Sp: 90.000000, Relative_Heading: 75.759311, Absolute Angle: 1.322105, error history: 20, history size: 20, time out time: -2147481531, error difference: 28.230736, over slew: 0, Actual_Vel1: 51.000000, Actual_Vel2: -77.200000, Actual_Vel3: 63.000000, Actual_Vel4: -66.800000
111: 27816 [INFO] CHASSIS_PID_TURN, Time: 2868, Actual_Vol1: 1971.000000, Actual_Vol2: -1725.000000, Actual_Vol3: 1435.000000, Actual_Vol4: -536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47425.872707, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 123.000000, position_r: -138.000000, Heading_Sp: 90.000000, Relative_Heading: 76.662947, Absolute Angle: 1.337877, error history: 20, history size: 20, time out time: -2147481531, error difference: 27.450849, over slew: 0, Actual_Vel1: 51.200000, Actual_Vel2: -70.600000, Actual_Vel3: 55.800000, Actual_Vel4: -61.600000
112: 27816 [INFO] CHASSIS_PID_TURN, Time: 2878, Actual_Vol1: 1823.000000, Actual_Vol2: -1558.000000, Actual_Vol3: 1398.000000, Actual_Vol4: -505.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47549.613356, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 123.000000, position_r: -141.000000, Heading_Sp: 90.000000, Relative_Heading: 77.625935, Absolute Angle: 1.354684, error history: 20, history size: 20, time out time: -2147481531, error difference: 26.739895, over slew: 0, Actual_Vel1: 48.400000, Actual_Vel2: -65.400000, Actual_Vel3: 53.400000, Actual_Vel4: -57.400000
113: 27816 [INFO] CHASSIS_PID_TURN, Time: 2888, Actual_Vol1: 2008.000000, Actual_Vol2: -1398.000000, Actual_Vol3: 1337.000000, Actual_Vol4: -437.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47665.214249, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 124.000000, position_r: -143.000000, Heading_Sp: 90.000000, Relative_Heading: 78.439911, Absolute Angle: 1.368890, error history: 20, history size: 20, time out time: -2147481531, error difference: 26.070933, over slew: 0, Actual_Vel1: 51.200000, Actual_Vel2: -59.600000, Actual_Vel3: 54.400000, Actual_Vel4: -54.600000
114: 27818 [INFO] CHASSIS_PID_TURN, Time: 2898, Actual_Vol1: 1983.000000, Actual_Vol2: -1312.000000, Actual_Vol3: 1318.000000, Actual_Vol4: -320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47773.174973, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 124.000000, position_r: -146.000000, Heading_Sp: 90.000000, Relative_Heading: 79.203928, Absolute Angle: 1.382225, error history: 20, history size: 20, time out time: -2147481531, error difference: 25.185245, over slew: 0, Actual_Vel1: 55.200000, Actual_Vel2: -55.000000, Actual_Vel3: 52.400000, Actual_Vel4: -56.600000
115: 27818 [INFO] CHASSIS_PID_TURN, Time: 2908, Actual_Vol1: 1756.000000, Actual_Vol2: -1220.000000, Actual_Vol3: 1238.000000, Actual_Vol4: -129.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47873.948486, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 125.000000, position_r: -148.000000, Heading_Sp: 90.000000, Relative_Heading: 79.922649, Absolute Angle: 1.394769, error history: 20, history size: 20, time out time: -2147481531, error difference: 24.339012, over slew: 0, Actual_Vel1: 53.000000, Actual_Vel2: -54.600000, Actual_Vel3: 53.000000, Actual_Vel4: -55.200000
116: 27818 [INFO] CHASSIS_PID_TURN, Time: 2918, Actual_Vol1: 1657.000000, Actual_Vol2: -1103.000000, Actual_Vol3: 1084.000000, Actual_Vol4: -18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47968.036170, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 126.000000, position_r: -148.000000, Heading_Sp: 90.000000, Relative_Heading: 80.591232, Absolute Angle: 1.406438, error history: 20, history size: 20, time out time: -2147481531, error difference: 23.058281, over slew: 0, Actual_Vel1: 47.200000, Actual_Vel2: -52.800000, Actual_Vel3: 50.800000, Actual_Vel4: -50.600000
117: 27820 [INFO] CHASSIS_PID_TURN, Time: 2928, Actual_Vol1: 1632.000000, Actual_Vol2: -942.000000, Actual_Vol3: 1023.000000, Actual_Vol4: -18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48057.897305, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 126.000000, position_r: -148.000000, Heading_Sp: 90.000000, Relative_Heading: 81.013886, Absolute Angle: 1.413815, error history: 20, history size: 20, time out time: -2147481531, error difference: 21.939389, over slew: 0, Actual_Vel1: 42.600000, Actual_Vel2: -48.800000, Actual_Vel3: 44.600000, Actual_Vel4: -42.000000
118: 27820 [INFO] CHASSIS_PID_TURN, Time: 2938, Actual_Vol1: 1626.000000, Actual_Vol2: -862.000000, Actual_Vol3: 992.000000, Actual_Vol4: -12.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48144.969708, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 128.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 81.292760, Absolute Angle: 1.418682, error history: 20, history size: 20, time out time: -2147481531, error difference: 20.684173, over slew: 0, Actual_Vel1: 40.400000, Actual_Vel2: -42.800000, Actual_Vel3: 39.400000, Actual_Vel4: -32.400000
119: 27820 [INFO] CHASSIS_PID_TURN, Time: 2948, Actual_Vol1: 1620.000000, Actual_Vol2: -918.000000, Actual_Vol3: 1023.000000, Actual_Vol4: -142.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48225.722402, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 128.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 81.924731, Absolute Angle: 1.429712, error history: 20, history size: 20, time out time: -2147481531, error difference: 19.803239, over slew: 0, Actual_Vel1: 38.200000, Actual_Vel2: -32.200000, Actual_Vel3: 37.600000, Actual_Vel4: -27.800000
120: 27822 [INFO] CHASSIS_PID_TURN, Time: 2958, Actual_Vol1: 1614.000000, Actual_Vol2: -1016.000000, Actual_Vol3: 1047.000000, Actual_Vol4: -154.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48304.506101, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 128.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.121630, Absolute Angle: 1.433149, error history: 20, history size: 20, time out time: -2147481531, error difference: 18.507555, over slew: 0, Actual_Vel1: 32.600000, Actual_Vel2: -27.800000, Actual_Vel3: 34.600000, Actual_Vel4: -26.000000
121: 27822 [INFO] CHASSIS_PID_TURN, Time: 2968, Actual_Vol1: 1657.000000, Actual_Vol2: -1060.000000, Actual_Vol3: 1090.000000, Actual_Vol4: -246.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48381.652920, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 129.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.285318, Absolute Angle: 1.436005, error history: 20, history size: 20, time out time: -2147481531, error difference: 17.100402, over slew: 0, Actual_Vel1: 32.600000, Actual_Vel2: -26.800000, Actual_Vel3: 30.200000, Actual_Vel4: -23.400000
122: 27822 [INFO] CHASSIS_PID_TURN, Time: 2978, Actual_Vol1: 1873.000000, Actual_Vol2: -1214.000000, Actual_Vol3: 1207.000000, Actual_Vol4: -339.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48456.512630, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 130.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.514029, Absolute Angle: 1.439997, error history: 20, history size: 20, time out time: -2147481531, error difference: 15.886748, over slew: 0, Actual_Vel1: 27.400000, Actual_Vel2: -28.800000, Actual_Vel3: 28.800000, Actual_Vel4: -22.800000
123: 27824 [INFO] CHASSIS_PID_TURN, Time: 2988, Actual_Vol1: 1990.000000, Actual_Vol2: -1275.000000, Actual_Vol3: 1257.000000, Actual_Vol4: -561.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48529.313264, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 130.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.719937, Absolute Angle: 1.443591, error history: 20, history size: 20, time out time: -2147481531, error difference: 14.672813, over slew: 0, Actual_Vel1: 27.400000, Actual_Vel2: -34.000000, Actual_Vel3: 25.600000, Actual_Vel4: -20.200000
124: 27824 [INFO] CHASSIS_PID_TURN, Time: 2998, Actual_Vol1: 2027.000000, Actual_Vol2: -1281.000000, Actual_Vol3: 1349.000000, Actual_Vol4: -788.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48601.212909, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 132.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.810036, Absolute Angle: 1.445164, error history: 20, history size: 20, time out time: -2147481531, error difference: 13.465240, over slew: 0, Actual_Vel1: 27.400000, Actual_Vel2: -35.800000, Actual_Vel3: 24.400000, Actual_Vel4: -19.000000
125: 27824 [INFO] CHASSIS_PID_TURN, Time: 3008, Actual_Vol1: 2107.000000, Actual_Vol2: -1287.000000, Actual_Vol3: 1441.000000, Actual_Vol4: -856.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48670.414490, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 132.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.079842, Absolute Angle: 1.449873, error history: 20, history size: 20, time out time: -2147481531, error difference: 11.653966, over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -34.200000, Actual_Vel3: 25.000000, Actual_Vel4: -19.200000
126: 27826 [INFO] CHASSIS_PID_TURN, Time: 3018, Actual_Vol1: 2242.000000, Actual_Vol2: -1380.000000, Actual_Vol3: 1540.000000, Actual_Vol4: -887.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48739.071279, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 133.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.134321, Absolute Angle: 1.450823, error history: 20, history size: 20, time out time: -2147481531, error difference: 10.537990, over slew: 0, Actual_Vel1: 23.000000, Actual_Vel2: -32.400000, Actual_Vel3: 26.400000, Actual_Vel4: -20.200000
127: 27826 [INFO] CHASSIS_PID_TURN, Time: 3028, Actual_Vol1: 2458.000000, Actual_Vol2: -1540.000000, Actual_Vol3: 1632.000000, Actual_Vol4: -998.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48806.355596, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 134.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.271568, Absolute Angle: 1.453219, error history: 20, history size: 20, time out time: -2147481531, error difference: 9.545978, over slew: 0, Actual_Vel1: 23.200000, Actual_Vel2: -27.200000, Actual_Vel3: 26.200000, Actual_Vel4: -29.000000
128: 27826 [INFO] CHASSIS_PID_TURN, Time: 3038, Actual_Vol1: 2526.000000, Actual_Vol2: -1571.000000, Actual_Vol3: 1756.000000, Actual_Vol4: -1115.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48872.371561, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 134.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.398404, Absolute Angle: 1.455432, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.586459, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -23.400000, Actual_Vel3: 25.000000, Actual_Vel4: -36.200000
129: 27828 [INFO] CHASSIS_PID_TURN, Time: 3048, Actual_Vol1: 2587.000000, Actual_Vol2: -1774.000000, Actual_Vol3: 1928.000000, Actual_Vol4: -1115.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48938.171963, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 135.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.419960, Absolute Angle: 1.455809, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.660669, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -23.400000, Actual_Vel3: 25.800000, Actual_Vel4: -45.400000
130: 27828 [INFO] CHASSIS_PID_TURN, Time: 3058, Actual_Vol1: 2667.000000, Actual_Vol2: -1885.000000, Actual_Vol3: 1990.000000, Actual_Vol4: -1109.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49002.811180, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 136.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.536078, Absolute Angle: 1.457835, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.873131, over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -23.400000, Actual_Vel3: 29.400000, Actual_Vel4: -47.600000
131: 27828 [INFO] CHASSIS_PID_TURN, Time: 3068, Actual_Vol1: 2747.000000, Actual_Vol2: -2273.000000, Actual_Vol3: 1996.000000, Actual_Vol4: -1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49066.288213, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 137.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.652297, Absolute Angle: 1.459864, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.026362, over slew: 0, Actual_Vel1: 27.800000, Actual_Vel2: -23.400000, Actual_Vel3: 28.200000, Actual_Vel4: -51.400000
132: 27830 [INFO] CHASSIS_PID_TURN, Time: 3078, Actual_Vol1: 2889.000000, Actual_Vol2: -2519.000000, Actual_Vol3: 2181.000000, Actual_Vol4: -1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49128.569187, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 138.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.771903, Absolute Angle: 1.461951, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.331992, over slew: 0, Actual_Vel1: 25.800000, Actual_Vel2: -23.400000, Actual_Vel3: 27.400000, Actual_Vel4: -50.600000
133: 27830 [INFO] CHASSIS_PID_TURN, Time: 3088, Actual_Vol1: 3043.000000, Actual_Vol2: -2815.000000, Actual_Vol3: 2445.000000, Actual_Vol4: -1109.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49189.569887, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 138.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.899930, Absolute Angle: 1.464186, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.696002, over slew: 0, Actual_Vel1: 26.200000, Actual_Vel2: -23.600000, Actual_Vel3: 28.400000, Actual_Vel4: -51.800000
134: 27830 [INFO] CHASSIS_PID_TURN, Time: 3098, Actual_Vol1: 3197.000000, Actual_Vol2: -3117.000000, Actual_Vol3: 2519.000000, Actual_Vol4: -1103.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49250.108965, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 140.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.946092, Absolute Angle: 1.464991, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.023444, over slew: 0, Actual_Vel1: 28.400000, Actual_Vel2: -23.600000, Actual_Vel3: 36.400000, Actual_Vel4: -52.800000
135: 27832 [INFO] CHASSIS_PID_TURN, Time: 3108, Actual_Vol1: 3308.000000, Actual_Vol2: -3326.000000, Actual_Vol3: 2532.000000, Actual_Vol4: -1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49308.070553, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 140.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.203841, Absolute Angle: 1.469490, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.612610, over slew: 0, Actual_Vel1: 28.400000, Actual_Vel2: -23.600000, Actual_Vel3: 46.200000, Actual_Vel4: -48.200000
136: 27832 [INFO] CHASSIS_PID_TURN, Time: 3118, Actual_Vol1: 3419.000000, Actual_Vol2: -3653.000000, Actual_Vol3: 2532.000000, Actual_Vol4: -1053.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49365.234823, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 141.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.283573, Absolute Angle: 1.470882, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.269686, over slew: 0, Actual_Vel1: 27.800000, Actual_Vel2: -19.000000, Actual_Vel3: 44.000000, Actual_Vel4: -43.400000
137: 27832 [INFO] CHASSIS_PID_TURN, Time: 3128, Actual_Vol1: 3511.000000, Actual_Vol2: -3979.000000, Actual_Vol3: 2624.000000, Actual_Vol4: -1109.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49420.458430, kD: 35.000000

0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 142.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.477639, Absolute Angle: 1.474269, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.18488
0, over slew: 0, Actual_Vel1: 30.200000, Actual_Vel2: -19.000000, Actual_Vel3: 33.200000, Actual_Vel4: -39.000000
138: 27834 [INFO] CHASSIS_PID_TURN, Time: 3138, Actual_Vol1: 3647.000000, Actual_Vol2: -4263.000000, Actual_Vol3: 2790.000000, Actual_Vol4: -1214.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49473.600384, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 143.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.685805, Absolute Angle: 1.477902, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.76107
4, over slew: 0, Actual_Vel1: 32.400000, Actual_Vel2: -19.000000, Actual_Vel3: 32.600000, Actual_Vel4: -30.600000
139: 27834 [INFO] CHASSIS_PID_TURN, Time: 3148, Actual_Vol1: 3776.000000, Actual_Vol2: -4466.000000, Actual_Vol3: 2852.000000, Actual_Vol4: -1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49524.480807, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 144.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.911958, Absolute Angle: 1.481849, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.79032
8, over slew: 0, Actual_Vel1: 34.600000, Actual_Vel2: -11.600000, Actual_Vel3: 39.600000, Actual_Vel4: -24.600000
140: 27834 [INFO] CHASSIS_PID_TURN, Time: 3158, Actual_Vol1: 3807.000000, Actual_Vol2: -4688.000000, Actual_Vol3: 2852.000000, Actual_Vol4: -1595.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49572.891563, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 145.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 85.158924, Absolute Angle: 1.486159, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.87360
6, over slew: 0, Actual_Vel1: 43.400000, Actual_Vel2: -9.600000, Actual_Vel3: 41.800000, Actual_Vel4: -20.600000
141: 27836 [INFO] CHASSIS_PID_TURN, Time: 3168, Actual_Vol1: 3819.000000, Actual_Vol2: -4897.000000, Actual_Vol3: 2889.000000, Actual_Vol4: -1669.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49618.651791, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 146.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 85.423977, Absolute Angle: 1.490785, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.90994
8, over slew: 0, Actual_Vel1: 44.600000, Actual_Vel2: -9.600000, Actual_Vel3: 41.200000, Actual_Vel4: -22.600000
142: 27836 [INFO] CHASSIS_PID_TURN, Time: 3178, Actual_Vol1: 3838.000000, Actual_Vol2: -5168.000000, Actual_Vol3: 2932.000000, Actual_Vol4: -1780.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49661.641180, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 148.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 85.701061, Absolute Angle: 1.495621, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.98112
4, over slew: 0, Actual_Vel1: 44.400000, Actual_Vel2: -9.600000, Actual_Vel3: 46.000000, Actual_Vel4: -33.000000
143: 27836 [INFO] CHASSIS_PID_TURN, Time: 3188, Actual_Vol1: 3942.000000, Actual_Vol2: -5341.000000, Actual_Vol3: 2932.000000, Actual_Vol4: -1885.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49700.753904, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 149.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 86.088728, Absolute Angle: 1.502387, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.27869
2, over slew: 0, Actual_Vel1: 48.600000, Actual_Vel2: -16.200000, Actual_Vel3: 50.800000, Actual_Vel4: -37.800000
144: 27838 [INFO] CHASSIS_PID_TURN, Time: 3198, Actual_Vol1: 3942.000000, Actual_Vol2: -5378.000000, Actual_Vol3: 2932.000000, Actual_Vol4: -1940.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49736.784071, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 150.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 86.396983, Absolute Angle: 1.507768, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.31714
1, over slew: 0, Actual_Vel1: 56.600000, Actual_Vel2: -21.800000, Actual_Vel3: 49.200000, Actual_Vel4: -41.000000
145: 27838 [INFO] CHASSIS_PID_TURN, Time: 3208, Actual_Vol1: 3702.000000, Actual_Vol2: -5513.000000, Actual_Vol3: 2901.000000, Actual_Vol4: -1947.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49769.467573, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 151.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 86.731650, Absolute Angle: 1.513609, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.59732
9, over slew: 0, Actual_Vel1: 57.800000, Actual_Vel2: -34.400000, Actual_Vel3: 49.200000, Actual_Vel4: -43.000000
146: 27838 [INFO] CHASSIS_PID_TURN, Time: 3218, Actual_Vol1: 3708.000000, Actual_Vol2: -5692.000000, Actual_Vol3: 2858.000000, Actual_Vol4: -1959.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49798.618332, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 152.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 87.084924, Absolute Angle: 1.519774, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.81335
6, over slew: 0, Actual_Vel1: 46.800000, Actual_Vel2: -31.200000, Actual_Vel3: 50.000000, Actual_Vel4: -47.000000
147: 27840 [INFO] CHASSIS_PID_TURN, Time: 3228, Actual_Vol1: 3788.000000, Actual_Vol2: -5760.000000, Actual_Vol3: 2778.000000, Actual_Vol4: -1829.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49824.108344, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 153.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 87.450999, Absolute Angle: 1.526164, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.05259
5, over slew: 0, Actual_Vel1: 52.200000, Actual_Vel2: -39.600000, Actual_Vel3: 51.200000, Actual_Vel4: -48.600000
148: 27840 [INFO] CHASSIS_PID_TURN, Time: 3238, Actual_Vol1: 3585.000000, Actual_Vol2: -5723.000000, Actual_Vol3: 2618.000000, Actual_Vol4: -1688.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49845.783434, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 154.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 87.832491, Absolute Angle: 1.532822, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.41253
1, over slew: 0, Actual_Vel1: 58.800000, Actual_Vel2: -40.400000, Actual_Vel3: 52.600000, Actual_Vel4: -50.600000
149: 27840 [INFO] CHASSIS_PID_TURN, Time: 3248, Actual_Vol1: 3456.000000, Actual_Vol2: -5723.000000, Actual_Vol3: 2569.000000, Actual_Vol4: -1669.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49863.436876, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 156.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 88.234656, Absolute Angle: 1.539841, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.69857
7, over slew: 0, Actual_Vel1: 49.000000, Actual_Vel2: -39.200000, Actual_Vel3: 56.200000, Actual_Vel4: -50.200000
150: 27842 [INFO] CHASSIS_PID_TURN, Time: 3258, Actual_Vol1: 3394.000000, Actual_Vol2: -5667.000000, Actual_Vol3: 2489.000000, Actual_Vol4: -1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49875.917217, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 157.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 88.751966, Absolute Angle: 1.548870, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.09966
9, over slew: 0, Actual_Vel1: 50.800000, Actual_Vel2: -40.800000, Actual_Vel3: 58.600000, Actual_Vel4: -50.400000
151: 27842 [INFO] CHASSIS_PID_TURN, Time: 3268, Actual_Vol1: 3197.000000, Actual_Vol2: -5544.000000, Actual_Vol3: 2372.000000, Actual_Vol4: -1368.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49884.109442, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 158.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 89.180778, Absolute Angle: 1.556354, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.40887
5, over slew: 0, Actual_Vel1: 52.600000, Actual_Vel2: -45.000000, Actual_Vel3: 60.600000, Actual_Vel4: -50.200000
152: 27842 [INFO] CHASSIS_PID_TURN, Time: 3278, Actual_Vol1: 3018.000000, Actual_Vol2: -5365.000000, Actual_Vol3: 2137.000000, Actual_Vol4: -1287.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49887.999861, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 158.000000, position_r: -151.000000, Heading_Sp: 90.000000, Relative_Heading: 89.610958, Absolute Angle: 1.563862, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.71102
8, over slew: 0, Actual_Vel1: 52.000000, Actual_Vel2: -46.600000, Actual_Vel3: 62.400000, Actual_Vel4: -47.800000
153: 27844 [INFO] CHASSIS_PID_TURN, Time: 3288, Actual_Vol1: 2969.000000, Actual_Vol2: -5267.000000, Actual_Vol3: 1953.000000, Actual_Vol4: -1238.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49887.538427, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 160.000000, position_r: -151.000000, Heading_Sp: 90.000000, Relative_Heading: 90.046143, Absolute Angle: 1.571457, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.10005
1, over slew: 0, Actual_Vel1: 52.600000, Actual_Vel2: -42.600000, Actual_Vel3: 58.400000, Actual_Vel4: -43.800000
154: 27844 [INFO] CHASSIS_PID_TURN, Time: 3298, Actual_Vol1: 2895.000000, Actual_Vol2: -5224.000000, Actual_Vol3: 1774.000000, Actual_Vol4: -1201.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49881.623834, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 161.000000, position_r: -151.000000, Heading_Sp: 90.000000, Relative_Heading: 90.591459, Absolute Angle: 1.580975, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.38761
8, over slew: 0, Actual_Vel1: 54.000000, Actual_Vel2: -44.800000, Actual_Vel3: 56.000000, Actual_Vel4: -43.800000
155: 27844 [INFO] CHASSIS_PID_TURN, Time: 3308, Actual_Vol1: 2784.000000, Actual_Vol2: -5156.000000, Actual_Vol3: 1602.000000, Actual_Vol4: -1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49871.295571, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 162.000000, position_r: -151.000000, Heading_Sp: 90.000000, Relative_Heading: 91.032826, Absolute Angle: 1.588678, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.74925
3, over slew: 0, Actual_Vel1: 52.800000, Actual_Vel2: -49.000000, Actual_Vel3: 53.400000, Actual_Vel4: -45.600000
156: 27846 [INFO] CHASSIS_PID_TURN, Time: 3318, Actual_Vol1: 2563.000000, Actual_Vol2: -5063.000000, Actual_Vol3: 1435.000000, Actual_Vol4: -1060.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49856.597414, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 163.000000, position_r: -152.000000, Heading_Sp: 90.000000, Relative_Heading: 91.469816, Absolute Angle: 1.596305, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.99217
6, over slew: 0, Actual_Vel1: 47.200000, Actual_Vel2: -47.200000, Actual_Vel3: 50.000000, Actual_Vel4: -46.800000
157: 27846 [INFO] CHASSIS_PID_TURN, Time: 3328, Actual_Vol1: 2519.000000, Actual_Vol2: -4891.000000, Actual_Vol3: 1306.000000, Actual_Vol4: -979.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49836.541888, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 164.000000, position_r: -153.000000, Heading_Sp: 90.000000, Relative_Heading: 92.005553, Absolute Angle: 1.605656, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.319748
, over slew: 0, Actual_Vel1: 44.800000, Actual_Vel2: -46.000000, Actual_Vel3: 47.800000, Actual_Vel4: -46.200000
158: 27846 [INFO] CHASSIS_PID_TURN, Time: 3338, Actual_Vol1: 2482.000000, Actual_Vol2: -4811.000000, Actual_Vol3: 1207.000000, Actual_Vol4: -893.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49811.233384, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 165.000000, position_r: -154.000000, Heading_Sp: 90.000000, Relative_Heading: 92.530850, Absolute Angle: 1.614824, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.618893
, over slew: 0, Actual_Vel1: 46.200000, Actual_Vel2: -51.400000, Actual_Vel3: 43.800000, Actual_Vel4: -45.600000
159: 27848 [INFO] CHASSIS_PID_TURN, Time: 3348, Actual_Vol1: 2322.000000, Actual_Vol2: -4700.000000, Actual_Vol3: 1146.000000, Actual_Vol4: -721.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49780.751273, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 165.000000, position_r: -154.000000, Heading_Sp: 90.000000, Relative_Heading: 93.048211, Absolute Angle: 1.623853, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.889287
, over slew: 0, Actual_Vel1: 44.000000, Actual_Vel2: -51.200000, Actual_Vel3: 40.600000, Actual_Vel4: -44.200000
160: 27848 [INFO] CHASSIS_PID_TURN, Time: 3358, Actual_Vol1: 2131.000000, Actual_Vol2: -4521.000000, Actual_Vol3: 1029.000000, Actual_Vol4: -567.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49747.189737, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 166.000000, position_r: -155.000000, Heading_Sp: 90.000000, Relative_Heading: 93.356154, Absolute Angle: 1.629228, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.932176
, over slew: 0, Actual_Vel1: 38.200000, Actual_Vel2: -48.600000, Actual_Vel3: 39.600000, Actual_Vel4: -38.800000
161: 27848 [INFO] CHASSIS_PID_TURN, Time: 3368, Actual_Vol1: 2107.000000, Actual_Vol2: -4337.000000, Actual_Vol3: 942.000000, Actual_Vol4: -487.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49708.553689, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 166.000000, position_r: -156.000000, Heading_Sp: 90.000000, Relative_Heading: 93.863605, Absolute Angle: 1.638085, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.162544,
over slew: 0, Actual_Vel1: 36.200000, Actual_Vel2: -45.400000, Actual_Vel3: 37.200000, Actual_Vel4: -34.000000
162: 27850 [INFO] CHASSIS_PID_TURN, Time: 3378, Actual_Vol1: 2088.000000, Actual_Vol2: -4115.000000, Actual_Vol3: 856.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49666.055912, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 166.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 94.249778, Absolute Angle: 1.644825, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.161050,
over slew: 0, Actual_Vel1: 36.800000, Actual_Vel2: -43.600000, Actual_Vel3: 35.000000, Actual_Vel4: -31.000000
163: 27850 [INFO] CHASSIS_PID_TURN, Time: 3388, Actual_Vol1: 1959.000000, Actual_Vol2: -3881.000000, Actual_Vol3: 696.000000, Actual_Vol4: -394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49619.829748, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 94.622616, Absolute Angle: 1.651332, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.225633,
over slew: 0, Actual_Vel1: 33.000000, Actual_Vel2: -41.600000, Actual_Vel3: 33.600000, Actual_Vel4: -24.600000
164: 27850 [INFO] CHASSIS_PID_TURN, Time: 3398, Actual_Vol1: 1663.000000, Actual_Vol2: -3782.000000, Actual_Vol3: 616.000000, Actual_Vol4: -283.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49569.077404, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 95.075234, Absolute Angle: 1.659232, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.343585,
over slew: 0, Actual_Vel1: 30.600000, Actual_Vel2: -39.600000, Actual_Vel3: 29.600000, Actual_Vel4: -21.400000
165: 27852 [INFO] CHASSIS_PID_TURN, Time: 3408, Actual_Vol1: 1546.000000, Actual_Vol2: -3560.000000, Actual_Vol3: 536.000000, Actual_Vol4: -234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49515.006829, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 95.407058, Absolute Angle: 1.665023, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.322133,
over slew: 0, Actual_Vel1: 29.200000, Actual_Vel2: -38.400000, Actual_Vel3: 25.600000, Actual_Vel4: -21.400000
166: 27852 [INFO] CHASSIS_PID_TURN, Time: 3418, Actual_Vol1: 1411.000000, Actual_Vol2: -3413.000000, Actual_Vol3: 480.000000, Actual_Vol4: -320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49457.745586, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 95.726124, Absolute Angle: 1.670592, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.275126,
over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -38.600000, Actual_Vel3: 24.600000, Actual_Vel4: -21.400000
167: 27852 [INFO] CHASSIS_PID_TURN, Time: 3428, Actual_Vol1: 1374.000000, Actual_Vol2: -3357.000000, Actual_Vol3: 419.000000, Actual_Vol4: -388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49398.431182, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 95.931440, Absolute Angle: 1.674175, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.098949,
over slew: 0, Actual_Vel1: 23.400000, Actual_Vel2: -32.200000, Actual_Vel3: 21.800000, Actual_Vel4: -21.400000
168: 27854 [INFO] CHASSIS_PID_TURN, Time: 3438, Actual_Vol1: 1306.000000, Actual_Vol2: -3283.000000, Actual_Vol3: 407.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49336.244616, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 96.218657, Absolute Angle: 1.679188, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.984001,
over slew: 0, Actual_Vel1: 20.800000, Actual_Vel2: -27.000000, Actual_Vel3: 20.200000, Actual_Vel4: -21.000000
169: 27854 [INFO] CHASSIS_PID_TURN, Time: 3448, Actual_Vol1: 1275.000000, Actual_Vol2: -3043.000000, Actual_Vol3: 265.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49272.297991, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 96.394662, Absolute Angle: 1.682260, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.642697,
over slew: 0, Actual_Vel1: 19.600000, Actual_Vel2: -27.200000, Actual_Vel3: 20.200000, Actual_Vel4: -21.000000
170: 27854 [INFO] CHASSIS_PID_TURN, Time: 3458, Actual_Vol1: 1220.000000, Actual_Vol2: -2907.000000, Actual_Vol3: 148.000000, Actual_Vol4: -444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49205.783535, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 96.651446, Absolute Angle: 1.686742, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.470668,
over slew: 0, Actual_Vel1: 19.600000, Actual_Vel2: -26.600000, Actual_Vel3: 18.200000, Actual_Vel4: -16.000000
171: 27856 [INFO] CHASSIS_PID_TURN, Time: 3468, Actual_Vol1: 1170.000000, Actual_Vol2: -2852.000000, Actual_Vol3: 111.000000, Actual_Vol4: -382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49136.865417, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 96.891812, Absolute Angle: 1.690937, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.280854,
over slew: 0, Actual_Vel1: 16.800000, Actual_Vel2: -26.600000, Actual_Vel3: 18.200000, Actual_Vel4: -16.000000
172: 27856 [INFO] CHASSIS_PID_TURN, Time: 3478, Actual_Vol1: 1164.000000, Actual_Vol2: -2834.000000, Actual_Vol3: 111.000000, Actual_Vol4: -400.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49066.720630, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 97.014479, Absolute Angle: 1.693078, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.968335,
over slew: 0, Actual_Vel1: 16.800000, Actual_Vel2: -21.800000, Actual_Vel3: 18.200000, Actual_Vel4: -12.400000
173: 27856 [INFO] CHASSIS_PID_TURN, Time: 3488, Actual_Vol1: 1146.000000, Actual_Vol2: -2784.000000, Actual_Vol3: 197.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48995.496073, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 97.122456, Absolute Angle: 1.694962, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.530996,
over slew: 0, Actual_Vel1: 16.800000, Actual_Vel2: -20.200000, Actual_Vel3: 18.200000, Actual_Vel4: -12.400000
174: 27858 [INFO] CHASSIS_PID_TURN, Time: 3498, Actual_Vol1: 1152.000000, Actual_Vol2: -2747.000000, Actual_Vol3: 290.000000, Actual_Vol4: -376.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48922.389438, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.310663, Absolute Angle: 1.698247, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.277837,
over slew: 0, Actual_Vel1: 16.800000, Actual_Vel2: -18.600000, Actual_Vel3: 18.200000, Actual_Vel4: -10.600000

175: 27858 [INFO] CHASSIS_PID_TURN, Time: 3508, Actual_Vol1: 1220.000000, Actual_Vol2: -2569.000000, Actual_Vol3: 308.000000, Actual_Vol4: -376.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48847.519744, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.486969, Absolute Angle: 1.701324, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.017154, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -18.600000, Actual_Vel3: 16.000000, Actual_Vel4: -10.600000

176: 27858 [INFO] CHASSIS_PID_TURN, Time: 3518, Actual_Vol1: 1250.000000, Actual_Vol2: -2655.000000, Actual_Vol3: 388.000000, Actual_Vol4: -382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48772.094316, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.542543, Absolute Angle: 1.702294, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.536990, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -18.600000, Actual_Vel3: 16.000000, Actual_Vel4: -10.600000

177: 27860 [INFO] CHASSIS_PID_TURN, Time: 3528, Actual_Vol1: 1257.000000, Actual_Vol2: -2717.000000, Actual_Vol3: 437.000000, Actual_Vol4: -265.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48696.299860, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.579446, Absolute Angle: 1.702938, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.048595, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -19.200000, Actual_Vel3: 13.000000, Actual_Vel4: -8.800000

178: 27860 [INFO] CHASSIS_PID_TURN, Time: 3538, Actual_Vol1: 1318.000000, Actual_Vol2: -2606.000000, Actual_Vol3: 419.000000, Actual_Vol4: -216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48620.253571, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.604629, Absolute Angle: 1.703378, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.556418, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -19.200000, Actual_Vel3: 13.000000, Actual_Vel4: -8.800000

179: 27860 [INFO] CHASSIS_PID_TURN, Time: 3548, Actual_Vol1: 1337.000000, Actual_Vol2: -2673.000000, Actual_Vol3: 419.000000, Actual_Vol4: -302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48544.124725, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.612885, Absolute Angle: 1.703522, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.256731, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -20.000000, Actual_Vel3: 10.400000, Actual_Vel4: -8.800000

180: 27862 [INFO] CHASSIS_PID_TURN, Time: 3558, Actual_Vol1: 1343.000000, Actual_Vol2: -2544.000000, Actual_Vol3: 394.000000, Actual_Vol4: -320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48468.107629, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.601710, Absolute Angle: 1.703327, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.749280, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -20.000000, Actual_Vel3: 10.400000, Actual_Vel4: -8.800000

181: 27862 [INFO] CHASSIS_PID_TURN, Time: 3568, Actual_Vol1: 1411.000000, Actual_Vol2: -2513.000000, Actual_Vol3: 314.000000, Actual_Vol4: -400.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48392.262733, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.584490, Absolute Angle: 1.703026, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.363107, over slew: 0, Actual_Vel1: 9.400000, Actual_Vel2: -20.000000, Actual_Vel3: 10.400000, Actual_Vel4: -8.400000

182: 27862 [INFO] CHASSIS_PID_TURN, Time: 3578, Actual_Vol1: 1435.000000, Actual_Vol2: -2649.000000, Actual_Vol3: 394.000000, Actual_Vol4: -444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48316.540486, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.572225, Absolute Angle: 1.702812, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.990268, over slew: 0, Actual_Vel1: 9.400000, Actual_Vel2: -20.000000, Actual_Vel3: 10.400000, Actual_Vel4: -8.400000

183: 27864 [INFO] CHASSIS_PID_TURN, Time: 3588, Actual_Vol1: 1540.000000, Actual_Vol2: -2717.000000, Actual_Vol3: 419.000000, Actual_Vol4: -480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48240.898340, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.564215, Absolute Angle: 1.702673, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.537650, over slew: 0, Actual_Vel1: 9.400000, Actual_Vel2: -17.600000, Actual_Vel3: 10.400000, Actual_Vel4: -8.400000

184: 27864 [INFO] CHASSIS_PID_TURN, Time: 3598, Actual_Vol1: 1565.000000, Actual_Vol2: -2710.000000, Actual_Vol3: 468.000000, Actual_Vol4: -505.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48165.358755, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.553959, Absolute Angle: 1.702494, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.205827, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -17.600000, Actual_Vel3: 9.600000, Actual_Vel4: -8.000000

185: 27864 [INFO] CHASSIS_PID_TURN, Time: 3608, Actual_Vol1: 1577.000000, Actual_Vol2: -2784.000000, Actual_Vol3: 511.000000, Actual_Vol4: -468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48089.902415, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.545634, Absolute Angle: 1.702348, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.886760, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -17.600000, Actual_Vel3: 9.600000, Actual_Vel4: -8.000000

186: 27866 [INFO] CHASSIS_PID_TURN, Time: 3618, Actual_Vol1: 1700.000000, Actual_Vol2: -2797.000000, Actual_Vol3: 517.000000, Actual_Vol4: -511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48014.533633, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.536878, Absolute Angle: 1.702195, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.681444, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -18.600000, Actual_Vel3: 9.600000, Actual_Vel4: -7.400000

187: 27866 [INFO] CHASSIS_PID_TURN, Time: 3628, Actual_Vol1: 1743.000000, Actual_Vol2: -2790.000000, Actual_Vol3: 561.000000, Actual_Vol4: -524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47939.225646, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.530799, Absolute Angle: 1.702089, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.394228, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -18.600000, Actual_Vel3: 9.600000, Actual_Vel4: -9.000000

188: 27866 [INFO] CHASSIS_PID_TURN, Time: 3638, Actual_Vol1: 1823.000000, Actual_Vol2: -2809.000000, Actual_Vol3: 598.000000, Actual_Vol4: -487.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47863.939426, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.528622, Absolute Angle: 1.702051, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.218222, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -18.600000, Actual_Vel3: 9.600000, Actual_Vel4: -9.000000

189: 27868 [INFO] CHASSIS_PID_TURN, Time: 3648, Actual_Vol1: 1860.000000, Actual_Vol2: -2883.000000, Actual_Vol3: 690.000000, Actual_Vol4: -474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47788.657116, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.528231, Absolute Angle: 1.702044, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.961439, over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: -11.600000, Actual_Vel3: 9.600000, Actual_Vel4: -9.000000

190: 27868 [INFO] CHASSIS_PID_TURN, Time: 3658, Actual_Vol1: 1866.000000, Actual_Vol2: -2889.000000, Actual_Vol3: 862.000000, Actual_Vol4: -480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47712.360770, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.629635, Absolute Angle: 1.703814, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.737823, over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: -11.600000, Actual_Vel3: 10.000000, Actual_Vel4: -8.800000

191: 27868 [INFO] CHASSIS_PID_TURN, Time: 3668, Actual_Vol1: 2027.000000, Actual_Vol2: -3037.000000, Actual_Vol3: 912.000000, Actual_Vol4: -468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47636.060306, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.630046, Absolute Angle: 1.703822, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.615568, over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: -11.600000, Actual_Vel3: 10.000000, Actual_Vel4: -11.200000

192: 27870 [INFO] CHASSIS_PID_TURN, Time: 3678, Actual_Vol1: 1940.000000, Actual_Vol2: -3166.000000, Actual_Vol3: 967.000000, Actual_Vol4: -462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47559.727195, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.633311, Absolute Angle: 1.703878, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.510855, over slew: 0, Actual_Vel1: 7.600000, Actual_Vel2: -0.000000, Actual_Vel3: 9.000000, Actual_Vel4: -11.200000

193: 27870 [INFO] CHASSIS_PID_TURN, Time: 3688, Actual_Vol1: 2014.000000, Actual_Vol2: -3265.000000, Actual_Vol3: 967.000000, Actual_Vol4: -474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47483.354396, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.637280, Absolute Angle: 1.703948, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.326616, over slew: 0, Actual_Vel1: 7.600000, Actual_Vel2: -0.000000, Actual_Vel3: 9.000000, Actual_Vel4: -12.400000

194: 27870 [INFO] CHASSIS_PID_TURN, Time: 3698, Actual_Vol1: 2057.000000, Actual_Vol2: -3289.000000, Actual_Vol3: 973.000000, Actual_Vol4: -450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47405.941914, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.741248, Absolute Angle: 1.705762, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.254279, over slew: 0, Actual_Vel1: 7.600000, Actual_Vel2: -0.000000, Actual_Vel3: 9.000000, Actual_Vel4: -12.800000

195: 27872 [INFO] CHASSIS_PID_TURN, Time: 3708, Actual_Vol1: 2070.000000, Actual_Vol2: -3302.000000, Actual_Vol3: 1016.000000, Actual_Vol4: -419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47328.432484, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.750943, Absolute Angle: 1.705932, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.222712, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 9.000000, Actual_Vel4: -14.000000

196: 27872 [INFO] CHASSIS_PID_TURN, Time: 3718, Actual_Vol1: 2070.000000, Actual_Vol2: -3363.000000, Actual_Vol3: 1010.000000, Actual_Vol4: -419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47250.835594, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.759689, Absolute Angle: 1.706084, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.231458, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 15.400000, Actual_Vel4: -14.000000

197: 27872 [INFO] CHASSIS_PID_TURN, Time: 3728, Actual_Vol1: 2070.000000, Actual_Vol2: -3382.000000, Actual_Vol3: 1016.000000, Actual_Vol4: -388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47173.122408, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.771319, Absolute Angle: 1.706287, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.243088, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 15.400000, Actual_Vel4: -13.400000

198: 27874 [INFO] CHASSIS_PID_TURN, Time: 3738, Actual_Vol1: 2070.000000, Actual_Vol2: -3610.000000, Actual_Vol3: 1053.000000, Actual_Vol4: -376.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47095.335467, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.778694, Absolute Angle: 1.706416, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.250463, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 15.400000, Actual_Vel4: -13.400000

199: 27874 [INFO] CHASSIS_PID_TURN, Time: 3748, Actual_Vol1: 2193.000000, Actual_Vol2: -3665.000000, Actual_Vol3: 1096.000000, Actual_Vol4: -370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47017.507757, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.782771, Absolute Angle: 1.706487, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.254540, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 15.400000, Actual_Vel4: -12.600000

200: 27874 [INFO] CHASSIS_PID_TURN, Time: 3758, Actual_Vol1: 2242.000000, Actual_Vol2: -3665.000000, Actual_Vol3: 1103.000000, Actual_Vol4: -363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46939.633743, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.787401, Absolute Angle: 1.706568, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.259170, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 19.000000, Actual_Vel4: -12.600000

201: 27876 [INFO] CHASSIS_PID_TURN, Time: 3768, Actual_Vol1: 2267.000000, Actual_Vol2: -3745.000000, Actual_Vol3: 1103.000000, Actual_Vol4: -246.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46861.719210, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.791453, Absolute Angle: 1.706639, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.263222, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -0.000000, Actual_Vel3: 19.000000, Actual_Vel4: -10.600000

202: 27876 [INFO] CHASSIS_PID_TURN, Time: 3778, Actual_Vol1: 2261.000000, Actual_Vol2: -3696.000000, Actual_Vol3: 1096.000000, Actual_Vol4: -203.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46783.717663, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.800155, Absolute Angle: 1.706790, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.271924, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -0.000000, Actual_Vel3: 19.000000, Actual_Vel4: -10.600000

203: 27876 [INFO] CHASSIS_PID_TURN, Time: 3788, Actual_Vol1: 2137.000000, Actual_Vol2: -3819.000000, Actual_Vol3: 1177.000000, Actual_Vol4: -314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46705.598880, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.811878, Absolute Angle: 1.706995, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.283647, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 19.000000, Actual_Vel4: -10.600000

204: 27878 [INFO] CHASSIS_PID_TURN, Time: 3798, Actual_Vol1: 2205.000000, Actual_Vol2: -3714.000000, Actual_Vol3: 1226.000000, Actual_Vol4: -407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46627.353858, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.824502, Absolute Angle: 1.707215, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.296271, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 19.000000, Actual_Vel4: -10.600000

205: 27878 [INFO] CHASSIS_PID_TURN, Time: 3808, Actual_Vol1: 2242.000000, Actual_Vol2: -3868.000000, Actual_Vol3: 1312.000000, Actual_Vol4: -462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46548.984715, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 97.836914, Absolute Angle: 1.707432, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.308683, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 46.800000, Actual_Vel4: -10.600000

206: 27878 [INFO] CHASSIS_PID_TURN, Time: 3818, Actual_Vol1: 2341.000000, Actual_Vol2: -3887.000000, Actual_Vol3: 1250.000000, Actual_Vol4: -456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46469.472234, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 97.951248, Absolute Angle: 1.709428, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.423017, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 46.800000, Actual_Vel4: -11.200000

207: 27880 [INFO] CHASSIS_PID_TURN, Time: 3828, Actual_Vol1: 2372.000000, Actual_Vol2: -3961.000000, Actual_Vol3: 1312.000000, Actual_Vol4: -450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46389.847372, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 97.962486, Absolute Angle: 1.709624, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.434255, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 20.600000, Actual_Vel4: -11.200000

208: 27880 [INFO] CHASSIS_PID_TURN, Time: 3838, Actual_Vol1: 2433.000000, Actual_Vol2: -3979.000000, Actual_Vol3: 1250.000000, Actual_Vol4: -468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46310.051927, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 97.979544, Absolute Angle: 1.709921, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.451313, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -11.600000, Actual_Vel3: 20.600000, Actual_Vel4: -11.200000

209: 27880 [INFO] CHASSIS_PID_TURN, Time: 3848, Actual_Vol1: 2458.000000, Actual_Vol2: -4115.000000, Actual_Vol3: 1244.000000, Actual_Vol4: -487.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46229.120042, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.093189, Absolute Angle: 1.711905, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.463554, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -11.600000, Actual_Vel3: 20.600000, Actual_Vel4: -18.200000

210: 27882 [INFO] CHASSIS_PID_TURN, Time: 3858, Actual_Vol1: 2464.000000, Actual_Vol2: -4121.000000, Actual_Vol3: 1312.000000, Actual_Vol4: -462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46148.036550, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.108349, Absolute Angle: 1.712169, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.478303, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -8.600000, Actual_Vel3: 20.600000, Actual_Vel4: -18.200000

211: 27882 [INFO] CHASSIS_PID_TURN, Time: 3868, Actual_Vol1: 2476.000000, Actual_Vol2: -4004.000000, Actual_Vol3: 1324.000000, Actual_Vol4: -480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46066.794142, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.124241, Absolute Angle: 1.712447, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.490930, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -8.600000, Actual_Vel3: 17.200000, Actual_Vel4: -18.200000

212: 27882 [INFO] CHASSIS_PID_TURN, Time: 3878, Actual_Vol1: 2470.000000, Actual_Vol2: -4109.000000, Actual_Vol3: 1404.000000, Actual_Vol4: -536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45985.363198, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.143094, Absolute Angle: 1.712776, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.505814

, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -8.600000, Actual_Vel3: 17.200000, Actual_Vel4: -34.600000
213: 27884 [INFO] CHASSIS_PID_TURN, Time: 3888, Actual_Vol1: 2581.000000, Actual_Vol2: -4127.000000, Actual_Vol3: 1417.000000, Actual_Vol4: -536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45903.733560, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.162964, Absolute Angle: 1.713123, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.421716, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -8.600000, Actual_Vel3: 17.200000, Actual_Vel4: -34.600000
214: 27884 [INFO] CHASSIS_PID_TURN, Time: 3898, Actual_Vol1: 2717.000000, Actual_Vol2: -4096.000000, Actual_Vol3: 1417.000000, Actual_Vol4: -511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45821.844177, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.188938, Absolute Angle: 1.713576, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.437995, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -12.600000, Actual_Vel3: 17.200000, Actual_Vel4: -28.000000
215: 27884 [INFO] CHASSIS_PID_TURN, Time: 3908, Actual_Vol1: 2760.000000, Actual_Vol2: -4146.000000, Actual_Vol3: 1540.000000, Actual_Vol4: -462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45739.650011, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.219417, Absolute Angle: 1.714108, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.459728, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -12.600000, Actual_Vel3: 17.200000, Actual_Vel4: -28.000000
216: 27886 [INFO] CHASSIS_PID_TURN, Time: 3918, Actual_Vol1: 2784.000000, Actual_Vol2: -4146.000000, Actual_Vol3: 1552.000000, Actual_Vol4: -474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45657.167732, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.248228, Absolute Angle: 1.714611, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.476909, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -12.600000, Actual_Vel3: 17.200000, Actual_Vel4: -28.000000
217: 27886 [INFO] CHASSIS_PID_TURN, Time: 3928, Actual_Vol1: 2556.000000, Actual_Vol2: -4158.000000, Actual_Vol3: 1571.000000, Actual_Vol4: -511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45573.456799, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.371093, Absolute Angle: 1.716755, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.592399, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -12.600000, Actual_Vel3: 8.000000, Actual_Vel4: -32.200000
218: 27886 [INFO] CHASSIS_PID_TURN, Time: 3938, Actual_Vol1: 2476.000000, Actual_Vol2: -4158.000000, Actual_Vol3: 1571.000000, Actual_Vol4: -474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45489.508852, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.394795, Absolute Angle: 1.717169, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.612024, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -12.600000, Actual_Vel3: 9.400000, Actual_Vel4: -29.800000
219: 27888 [INFO] CHASSIS_PID_TURN, Time: 3948, Actual_Vol1: 2470.000000, Actual_Vol2: -4158.000000, Actual_Vol3: 1448.000000, Actual_Vol4: -468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45405.290958, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.421789, Absolute Angle: 1.717640, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.634388, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.000000, Actual_Vel3: 9.400000, Actual_Vel4: -17.600000
220: 27888 [INFO] CHASSIS_PID_TURN, Time: 3958, Actual_Vol1: 2587.000000, Actual_Vol2: -4152.000000, Actual_Vol3: 1423.000000, Actual_Vol4: -456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45319.727269, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.556369, Absolute Angle: 1.719989, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.764916, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.000000, Actual_Vel3: 9.400000, Actual_Vel4: -17.600000
221: 27888 [INFO] CHASSIS_PID_TURN, Time: 3968, Actual_Vol1: 2717.000000, Actual_Vol2: -4139.000000, Actual_Vol3: 1540.000000, Actual_Vol4: -450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45233.833637, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.589363, Absolute Angle: 1.720565, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.789208, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.000000, Actual_Vel3: 9.400000, Actual_Vel4: -17.600000
222: 27890 [INFO] CHASSIS_PID_TURN, Time: 3978, Actual_Vol1: 2753.000000, Actual_Vol2: -4152.000000, Actual_Vol3: 1429.000000, Actual_Vol4: -456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45147.635518, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.619812, Absolute Angle: 1.721096, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.807934, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.200000, Actual_Vel3: 12.400000, Actual_Vel4: -17.600000
223: 27890 [INFO] CHASSIS_PID_TURN, Time: 3988, Actual_Vol1: 2821.000000, Actual_Vol2: -4152.000000, Actual_Vol3: 1417.000000, Actual_Vol4: -480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45061.136110, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.649941, Absolute Angle: 1.721622, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.825439, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.200000, Actual_Vel3: 12.400000, Actual_Vel4: -17.600000
224: 27890 [INFO] CHASSIS_PID_TURN, Time: 3998, Actual_Vol1: 2852.000000, Actual_Vol2: -4158.000000, Actual_Vol3: 1349.000000, Actual_Vol4: -524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44974.370561, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.676555, Absolute Angle: 1.722087, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.839641, over slew: 0, Actual_Vel1: 21.000000, Actual_Vel2: -11.200000, Actual_Vel3: 11.600000, Actual_Vel4: -17.600000
225: 27892 [INFO] CHASSIS_PID_TURN, Time: 4008, Actual_Vol1: 2587.000000, Actual_Vol2: -4176.000000, Actual_Vol3: 1411.000000, Actual_Vol4: -536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44887.340193, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.703037, Absolute Angle: 1.722549, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.751789, over slew: 0, Actual_Vel1: 21.000000, Actual_Vel2: -11.200000, Actual_Vel3: 11.600000, Actual_Vel4: -17.600000
226: 27892 [INFO] CHASSIS_PID_TURN, Time: 4018, Actual_Vol1: 2495.000000, Actual_Vol2: -4183.000000, Actual_Vol3: 1423.000000, Actual_Vol4: -524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44800.018468, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.732172, Absolute Angle: 1.723057, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.769686, over slew: 0, Actual_Vel1: 21.000000, Actual_Vel2: -11.200000, Actual_Vel3: 12.800000, Actual_Vel4: -17.600000
227: 27892 [INFO] CHASSIS_PID_TURN, Time: 4028, Actual_Vol1: 2612.000000, Actual_Vol2: -4170.000000, Actual_Vol3: 1343.000000, Actual_Vol4: -591.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44712.452141, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.756633, Absolute Angle: 1.723484, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.777088, over slew: 0, Actual_Vel1: 21.000000, Actual_Vel2: -11.200000, Actual_Vel3: 12.800000, Actual_Vel4: -17.600000
228: 27894 [INFO] CHASSIS_PID_TURN, Time: 4038, Actual_Vol1: 2741.000000, Actual_Vol2: -4164.000000, Actual_Vol3: 1324.000000, Actual_Vol4: -678.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44624.665178, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.778696, Absolute Angle: 1.723869, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.685508, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -7.800000, Actual_Vel3: 12.800000, Actual_Vel4: -17.600000
229: 27894 [INFO] CHASSIS_PID_TURN, Time: 4048, Actual_Vol1: 2544.000000, Actual_Vol2: -4275.000000, Actual_Vol3: 1331.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44536.591306, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.807387, Absolute Angle: 1.724370, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.699038, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -7.800000, Actual_Vel3: 15.600000, Actual_Vel4: -10.200000
230: 27894 [INFO] CHASSIS_PID_TURN, Time: 4058, Actual_Vol1: 2482.000000, Actual_Vol2: -4293.000000, Actual_Vol3: 1337.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44447.304447, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928686, Absolute Angle: 1.726487, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.804445, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -7.800000, Actual_Vel3: 15.600000, Actual_Vel4: -10.200000
231: 27896 [INFO] CHASSIS_PID_TURN, Time: 4068, Actual_Vol1: 2470.000000, Actual_Vol2: -4275.000000, Actual_Vol3: 1331.000000, Actual_Vol4: -647.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44357.807529, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.949692, Absolute Angle: 1.726854, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.806597, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -7.000000, Actual_Vel3: 15.600000, Actual_Vel4: -11.000000
232: 27896 [INFO] CHASSIS_PID_TURN, Time: 4078, Actual_Vol1: 764.000000, Actual_Vol2: -1318.000000, Actual_Vol3: 296.000000, Actual_Vol4: -117.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44268.144489, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.966304, Absolute Angle: 1.727144, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.803340, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: 17.600000, Actual_Vel3: 15.600000, Actual_Vel4: -12.400000
233: 27896 [INFO] CHASSIS_PID_TURN, Time: 4088, Actual_Vol1: 203.000000, Actual_Vol2: -425.000000, Actual_Vol3: 62.000000, Actual_Vol4: -18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44178.369090, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.977540, Absolute Angle: 1.727340, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.788602, over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: 17.600000, Actual_Vel3: 15.600000, Actual_Vel4: -12.400000
234: 27898 [INFO] CHASSIS_PID_TURN, Time: 4098, Actual_Vol1: 117.000000, Actual_Vol2: -228.000000, Actual_Vol3: 80.000000, Actual_Vol4: -259.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44088.510075, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.985901, Absolute Angle: 1.727486, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.766485, over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: 16.800000, Actual_Vel3: 15.600000, Actual_Vel4: -4.000000
235: 27898 [INFO] CHASSIS_PID_TURN, Time: 4108, Actual_Vol1: 105.000000, Actual_Vol2: -179.000000, Actual_Vol3: 271.000000, Actual_Vol4: -407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43998.622738, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.988734, Absolute Angle: 1.727535, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.740506, over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: 16.800000, Actual_Vel3: 13.000000, Actual_Vel4: -4.000000
236: 27898 [INFO] CHASSIS_PID_TURN, Time: 4118, Actual_Vol1: 105.000000, Actual_Vol2: -154.000000, Actual_Vol3: 302.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43908.789195, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.983354, Absolute Angle: 1.727441, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.617640, over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: 16.800000, Actual_Vel3: 13.000000, Actual_Vel4: -4.000000
237: 27900 [INFO] CHASSIS_PID_TURN, Time: 4128, Actual_Vol1: 92.000000, Actual_Vol2: -142.000000, Actual_Vol3: 302.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43819.109721, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.967947, Absolute Angle: 1.727172, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.593939, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 16.800000, Actual_Vel3: 13.000000, Actual_Vel4: -4.000000
238: 27900 [INFO] CHASSIS_PID_TURN, Time: 4138, Actual_Vol1: 92.000000, Actual_Vol2: -142.000000, Actual_Vol3: 302.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43729.547062, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956266, Absolute Angle: 1.726968, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.566944, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 16.800000, Actual_Vel3: 15.400000, Actual_Vel4: -4.000000
239: 27900 [INFO] CHASSIS_PID_TURN, Time: 4148, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 308.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43640.064256, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.948281, Absolute Angle: 1.726829, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.432365, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 16.800000, Actual_Vel3: 15.400000, Actual_Vel4: -4.000000
240: 27902 [INFO] CHASSIS_PID_TURN, Time: 4158, Actual_Vol1: 105.000000, Actual_Vol2: -142.000000, Actual_Vol3: 302.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43550.586438, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.947782, Absolute Angle: 1.726820, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.399371, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.800000, Actual_Vel3: 15.400000, Actual_Vel4: -4.000000
241: 27902 [INFO] CHASSIS_PID_TURN, Time: 4168, Actual_Vol1: 86.000000, Actual_Vol2: -142.000000, Actual_Vol3: 302.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43461.105789, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.948065, Absolute Angle: 1.726825, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.368922, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.800000, Actual_Vel3: 77.000000, Actual_Vel4: 0.000000
242: 27902 [INFO] CHASSIS_PID_TURN, Time: 4178, Actual_Vol1: 86.000000, Actual_Vol2: -129.000000, Actual_Vol3: 308.000000, Actual_Vol4: -444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43371.613608, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.949218, Absolute Angle: 1.726845, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.338793, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.800000, Actual_Vel3: 77.000000, Actual_Vel4: 0.000000
243: 27904 [INFO] CHASSIS_PID_TURN, Time: 4188, Actual_Vol1: 99.000000, Actual_Vol2: -136.000000, Actual_Vol3: 308.000000, Actual_Vol4: -326.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43282.054588, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.955902, Absolute Angle: 1.726962, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.312179, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.200000, Actual_Vel3: 77.000000, Actual_Vel4: 0.000000
244: 27904 [INFO] CHASSIS_PID_TURN, Time: 4198, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 302.000000, Actual_Vol4: -314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43192.466032, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958856, Absolute Angle: 1.727014, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.285697, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
245: 27904 [INFO] CHASSIS_PID_TURN, Time: 4208, Actual_Vol1: 86.000000, Actual_Vol2: -136.000000, Actual_Vol3: 308.000000, Actual_Vol4: -308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43102.879321, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958671, Absolute Angle: 1.727010, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.256561, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
246: 27906 [INFO] CHASSIS_PID_TURN, Time: 4218, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 413.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43013.286791, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.959253, Absolute Angle: 1.727021, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.232101, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
247: 27906 [INFO] CHASSIS_PID_TURN, Time: 4228, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 431.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42923.707792, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957900, Absolute Angle: 1.726997, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.210037, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
248: 27906 [INFO] CHASSIS_PID_TURN, Time: 4238, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 431.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42834.101375, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960642, Absolute Angle: 1.727045, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.181347, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
249: 27908 [INFO] CHASSIS_PID_TURN, Time: 4248, Actual_Vol1: 86.000000, Actual_Vol2: -129.000000, Actual_Vol3: 431.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42744.553756, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.954762, Absolute Angle: 1.726942, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.060048, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
250: 27908 [INFO] CHASSIS_PID_TURN, Time: 4258, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 425.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42654.974120, kD: 35.000000, kI:

0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957964, Absolute Angle: 1.726998, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000

251: 27908 [INFO] CHASSIS_PID_TURN, Time: 4268, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 431.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42565.377039, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.959708, Absolute Angle: 1.727028, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000

252: 27910 [INFO] CHASSIS_PID_TURN, Time: 4278, Actual_Vol1: 191.000000, Actual_Vol2: -216.000000, Actual_Vol3: 499.000000, Actual_Vol4: -505.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42475.774832, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960221, Absolute Angle: 1.727037, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000

253: 27910 [INFO] CHASSIS_PID_TURN, Time: 4288, Actual_Vol1: 222.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42386.182342, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.959249, Absolute Angle: 1.727020, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000

254: 27910 [INFO] CHASSIS_PID_TURN, Time: 4298, Actual_Vol1: 228.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42296.609836, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957251, Absolute Angle: 1.726986, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000

255: 27912 [INFO] CHASSIS_PID_TURN, Time: 4308, Actual_Vol1: 222.000000, Actual_Vol2: -240.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42207.050096, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.955974, Absolute Angle: 1.726963, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035572, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000

256: 27912 [INFO] CHASSIS_PID_TURN, Time: 4318, Actual_Vol1: 228.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42117.481242, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956885, Absolute Angle: 1.726979, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020166, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

257: 27912 [INFO] CHASSIS_PID_TURN, Time: 4328, Actual_Vol1: 222.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42027.942363, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953888, Absolute Angle: 1.726927, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012860, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

258: 27914 [INFO] CHASSIS_PID_TURN, Time: 4338, Actual_Vol1: 228.000000, Actual_Vol2: -246.000000, Actual_Vol3: 524.000000, Actual_Vol4: -511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41938.398771, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.954359, Absolute Angle: 1.726935, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012860, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

259: 27914 [INFO] CHASSIS_PID_TURN, Time: 4348, Actual_Vol1: 228.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41848.864117, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953465, Absolute Angle: 1.726920, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012860, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

260: 27914 [INFO] CHASSIS_PID_TURN, Time: 4358, Actual_Vol1: 302.000000, Actual_Vol2: -308.000000, Actual_Vol3: 598.000000, Actual_Vol4: -591.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41759.300926, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956319, Absolute Angle: 1.726969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012577, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

261: 27916 [INFO] CHASSIS_PID_TURN, Time: 4368, Actual_Vol1: 308.000000, Actual_Vol2: -326.000000, Actual_Vol3: 610.000000, Actual_Vol4: -616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41669.698142, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960278, Absolute Angle: 1.727038, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011424, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

262: 27916 [INFO] CHASSIS_PID_TURN, Time: 4378, Actual_Vol1: 308.000000, Actual_Vol2: -370.000000, Actual_Vol3: 622.000000, Actual_Vol4: -604.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41580.074952, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962319, Absolute Angle: 1.727074, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008854, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

263: 27916 [INFO] CHASSIS_PID_TURN, Time: 4388, Actual_Vol1: 308.000000, Actual_Vol2: -326.000000, Actual_Vol3: 616.000000, Actual_Vol4: -610.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41490.451738, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962321, Absolute Angle: 1.727074, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008856, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

264: 27918 [INFO] CHASSIS_PID_TURN, Time: 4398, Actual_Vol1: 308.000000, Actual_Vol2: -333.000000, Actual_Vol3: 616.000000, Actual_Vol4: -610.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41400.810398, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.964134, Absolute Angle: 1.727106, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.010669, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

265: 27918 [INFO] CHASSIS_PID_TURN, Time: 4408, Actual_Vol1: 314.000000, Actual_Vol2: -326.000000, Actual_Vol3: 616.000000, Actual_Vol4: -616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41311.139810, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.967059, Absolute Angle: 1.727157, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

266: 27918 [INFO] CHASSIS_PID_TURN, Time: 4418, Actual_Vol1: 308.000000, Actual_Vol2: -326.000000, Actual_Vol3: 622.000000, Actual_Vol4: -616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41221.531368, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960844, Absolute Angle: 1.727048, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

267: 27920 [INFO] CHASSIS_PID_TURN, Time: 4428, Actual_Vol1: 314.000000, Actual_Vol2: -333.000000, Actual_Vol3: 622.000000, Actual_Vol4: -616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41131.927400, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960397, Absolute Angle: 1.727041, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

268: 27920 [INFO] CHASSIS_PID_TURN, Time: 4438, Actual_Vol1: 314.000000, Actual_Vol2: -363.000000, Actual_Vol3: 610.000000, Actual_Vol4: -622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41042.356648, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957075, Absolute Angle: 1.726983, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

269: 27920 [INFO] CHASSIS_PID_TURN, Time: 4448, Actual_Vol1: 388.000000, Actual_Vol2: -333.000000, Actual_Vol3: 610.000000, Actual_Vol4: -604.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40952.777543, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957910, Absolute Angle: 1.726997, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

270: 27922 [INFO] CHASSIS_PID_TURN, Time: 4458, Actual_Vol1: 425.000000, Actual_Vol2: -431.000000, Actual_Vol3: 690.000000, Actual_Vol4: -653.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40863.135917, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.964163, Absolute Angle: 1.727106, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

271: 27922 [INFO] CHASSIS_PID_TURN, Time: 4468, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 702.000000, Actual_Vol4: -702.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40773.476558, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.965936, Absolute Angle: 1.727137, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

272: 27922 [INFO] CHASSIS_PID_TURN, Time: 4478, Actual_Vol1: 431.000000, Actual_Vol2: -444.000000, Actual_Vol3: 708.000000, Actual_Vol4: -702.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40683.805494, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.967106, Absolute Angle: 1.727158, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013641, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

273: 27924 [INFO] CHASSIS_PID_TURN, Time: 4488, Actual_Vol1: 437.000000, Actual_Vol2: -450.000000, Actual_Vol3: 702.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40594.176729, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962876, Absolute Angle: 1.727084, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013641, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

274: 27924 [INFO] CHASSIS_PID_TURN, Time: 4498, Actual_Vol1: 437.000000, Actual_Vol2: -444.000000, Actual_Vol3: 708.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40504.473495, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.970323, Absolute Angle: 1.727214, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

275: 27924 [INFO] CHASSIS_PID_TURN, Time: 4508, Actual_Vol1: 431.000000, Actual_Vol2: -456.000000, Actual_Vol3: 708.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40414.745208, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.972829, Absolute Angle: 1.727257, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019363, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

276: 27926 [INFO] CHASSIS_PID_TURN, Time: 4518, Actual_Vol1: 437.000000, Actual_Vol2: -444.000000, Actual_Vol3: 708.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40325.055298, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.968991, Absolute Angle: 1.727191, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019363, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

277: 27926 [INFO] CHASSIS_PID_TURN, Time: 4528, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 715.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40235.343287, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.971201, Absolute Angle: 1.727229, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019363, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

278: 27926 [INFO] CHASSIS_PID_TURN, Time: 4538, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 702.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40145.617896, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.972539, Absolute Angle: 1.727252, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019363, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

279: 27928 [INFO] CHASSIS_PID_TURN, Time: 4548, Actual_Vol1: 437.000000, Actual_Vol2: -450.000000, Actual_Vol3: 708.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40055.909833, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.970806, Absolute Angle: 1.727222, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016510, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

280: 27928 [INFO] CHASSIS_PID_TURN, Time: 4558, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 708.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39966.260268, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.964956, Absolute Angle: 1.727120, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

281: 27928 [INFO] CHASSIS_PID_TURN, Time: 4568, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 702.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39876.647964, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.961230, Absolute Angle: 1.727055, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

282: 27930 [INFO] CHASSIS_PID_TURN, Time: 4578, Actual_Vol1: 431.000000, Actual_Vol2: -456.000000, Actual_Vol3: 708.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39787.018388, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962958, Absolute Angle: 1.727085, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

283: 27930 [INFO] CHASSIS_PID_TURN, Time: 4588, Actual_Vol1: 431.000000, Actual_Vol2: -456.000000, Actual_Vol3: 702.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39697.403715, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.961467, Absolute Angle: 1.727059, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

284: 27930 [INFO] CHASSIS_PID_TURN, Time: 4598, Actual_Vol1: 92.000000, Actual_Vol2: -105.000000, Actual_Vol3: 136.000000, Actual_Vol4: -123.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39607.754314, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.964940, Absolute Angle: 1.727120, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

285: 27932 [INFO] CHASSIS_PID_TURN, Time: 4608, Actual_Vol1: 18.000000, Actual_Vol2: -25.000000, Actual_Vol3: 31.000000, Actual_Vol4: -18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39518.044655, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.970966, Absolute Angle: 1.727225, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

286: 27932 [INFO] CHASSIS_PID_TURN, Time: 4618, Actual_Vol1: 0.000000, Actual_Vol2: -12.000000, Actual_Vol3: 12.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39428.331340, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.971332, Absolute Angle: 1.727231, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

287: 27932 [INFO] CHASSIS_PID_TURN, Time: 4628, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39338.579359, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.975198, Absolute Angle: 1.727299, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018123, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

288: 27934 [INFO] CHASSIS_PID_TURN, Time: 4638, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39248.876868, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.970249, Absolute Angle: 1.727212, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017288, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

289: 27934 [INFO] CHASSIS_PID_TURN, Time: 4648, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39159.193703, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.968317, Absolute Angle: 1.727179, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013968, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

290: 27934 [INFO] CHASSIS_PID_TURN, Time: 4658, Actual_Vol1: -6.000000, Actual_Vol2: 0.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39069.539490, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.965421, Absolute Angle: 1.727128, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013968, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

291: 27936 [INFO] CHASSIS_PID_TURN, Time: 4668, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38979.882424, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.965707, Absolute Angle: 1.727133, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013968, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

292: 27936 [INFO] CHASSIS_PID_TURN, Time: 4678, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38890.263209, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.961921, Absolute Angle: 1.727067, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013968, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

293: 27936 [INFO] CHASSIS_PID_TURN, Time: 4688, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38800.688767, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957444, Absolute Angle: 1.726989, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017754, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

294: 27938 [INFO] CHASSIS_PID_TURN, Time: 4698, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38711.135370, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.955340, Absolute Angle: 1.726952, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

295: 27938 [INFO] CHASSIS_PID_TURN, Time: 4708, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38621.513387, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962198, Absolute Angle: 1.727072, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

296: 27938 [INFO] CHASSIS_PID_TURN, Time: 4718, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38531.900666, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.961272, Absolute Angle: 1.727056, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

297: 27940 [INFO] CHASSIS_PID_TURN, Time: 4728, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38442.322121, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957855, Absolute Angle: 1.726996, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

298: 27940 [INFO] CHASSIS_PID_TURN, Time: 4738, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38352.749563, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957256, Absolute Angle: 1.726986, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

299: 27940 [INFO] CHASSIS_PID_TURN, Time: 4748, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: -6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38263.190770, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.955879, Absolute Angle: 1.726962, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

300: 27942 [INFO] CHASSIS_PID_TURN, Time: 4758, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38173.615329, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957544, Absolute Angle: 1.726991, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

301: 27942 [INFO] CHASSIS_PID_TURN, Time: 4768, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38084.020793, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.959454, Absolute Angle: 1.727024, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

302: 27942 [INFO] CHASSIS_PID_TURN, Time: 4778, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37994.416885, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960391, Absolute Angle: 1.727040, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

303: 27944 [INFO] CHASSIS_PID_TURN, Time: 4788, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37904.878121, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953876, Absolute Angle: 1.726927, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021322, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000

304: 27944 [INFO] CHASSIS_PID_TURN, Time: 4798, Actual_Vol1: 6.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37815.349362, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.952876, Absolute Angle: 1.726909, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022322, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

305: 27944 [INFO] CHASSIS_PID_TURN, Time: 4808, Actual_Vol1: -6.000000, Actual_Vol2: 0.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37725.779093, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957027, Absolute Angle: 1.726982, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022322, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

306: 27946 [INFO] CHASSIS_PID_TURN, Time: 4818, Actual_Vol1: -6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37636.208280, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957081, Absolute Angle: 1.726983, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022322, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

307: 27946 [INFO] CHASSIS_PID_TURN, Time: 4828, Actual_Vol1: 0.000000, Actual_Vol2: 6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37546.676796, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953148, Absolute Angle: 1.726914, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017373, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

308: 27946 [INFO] CHASSIS_PID_TURN, Time: 4838, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37457.204253, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.947254, Absolute Angle: 1.726811, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021062, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

309: 27948 [INFO] CHASSIS_PID_TURN, Time: 4848, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37367.680406, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.952385, Absolute Angle: 1.726901, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018452, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

310: 27948 [INFO] CHASSIS_PID_TURN, Time: 4858, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37278.144835, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953557, Absolute Angle: 1.726921, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018452, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

311: 27948 [INFO] CHASSIS_PID_TURN, Time: 4868, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37188.561212, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958362, Absolute Angle: 1.727005, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014944, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

312: 27950 [INFO] CHASSIS_PID_TURN, Time: 4878, Actual_Vol1: 6.000000, Actual_Vol2: 0.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37098.921311, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.963990, Absolute Angle: 1.727103, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

313: 27950 [INFO] CHASSIS_PID_TURN, Time: 4888, Actual_Vol1: -6.000000, Actual_Vol2: 62.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37009.340856, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958046, Absolute Angle: 1.726999, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

314: 27950 [INFO] CHASSIS_PID_TURN, Time: 4898, Actual_Vol1: -80.000000, Actual_Vol2: 62.000000, Actual_Vol3: 6.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36919.752900, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958796, Absolute Angle: 1.727013, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

315: 27952 [INFO] CHASSIS_PID_TURN, Time: 4908, Actual_Vol1: -18.000000, Actual_Vol2: 12.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36830.127824, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962508, Absolute Angle: 1.727077, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

316: 27952 [INFO] CHASSIS_PID_TURN, Time: 4918, Actual_Vol1: 0.000000, Actual_Vol2: 68.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36740.541347, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958648, Absolute Angle: 1.727010, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

317: 27952 [INFO] CHASSIS_PID_TURN, Time: 4928, Actual_Vol1: -74.000000, Actual_Vol2: 209.000000, Actual_Vol3: 0.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36650.976928, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956442, Absolute Angle: 1.726971, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

318: 27954 [INFO] CHASSIS_PID_TURN, Time: 4938, Actual_Vol1: -277.000000, Actual_Vol2: 320.000000, Actual_Vol3: -6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36561.409907, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956702, Absolute Angle: 1.726976, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

319: 27954 [INFO] CHASSIS_PID_TURN, Time: 4948, Actual_Vol1: -142.000000, Actual_Vol2: 370.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36471.829030, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958088, Absolute Angle: 1.727000, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

320: 27954 [INFO] CHASSIS_PID_TURN, Time: 4958, Actual_Vol1: -105.000000, Actual_Vol2: 376.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36382.268935, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956009, Absolute Angle: 1.726964, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

321: 27956 [INFO] CHASSIS_PID_TURN, Time: 4968, Actual_Vol1: -99.000000, Actual_Vol2: 382.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36292.779706, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.948923, Absolute Angle: 1.726840, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

322: 27956 [INFO] CHASSIS_PID_TURN, Time: 4978, Actual_Vol1: -99.000000, Actual_Vol2: 370.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36203.318032, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.946167, Absolute Angle: 1.726792, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017823, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

323: 27956 [INFO] CHASSIS_PID_TURN, Time: 4988, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36113.907236, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.941080, Absolute Angle: 1.726703, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022910, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

324: 27958 [INFO] CHASSIS_PID_TURN, Time: 4998, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36024.521528, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938571, Absolute Angle: 1.726660, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025419, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

325: 27958 [INFO] CHASSIS_PID_TURN, Time: 5008, Actual_Vol1: -92.000000, Actual_Vol2: 326.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35935.199594, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.932193, Absolute Angle: 1.726548, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031797, over sle

w: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
326: 27958 [INFO] CHASSIS_PID_TURN, Time: 5018, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35845.904096, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.929550, Absolute Angle: 1.726502, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.034440, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
327: 27960 [INFO] CHASSIS_PID_TURN, Time: 5028, Actual_Vol1: -86.000000, Actual_Vol2: 320.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35756.633572, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.927052, Absolute Angle: 1.726459, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
328: 27960 [INFO] CHASSIS_PID_TURN, Time: 5038, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35667.345778, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928779, Absolute Angle: 1.726489, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
329: 27960 [INFO] CHASSIS_PID_TURN, Time: 5048, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35578.064756, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928102, Absolute Angle: 1.726477, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
330: 27962 [INFO] CHASSIS_PID_TURN, Time: 5058, Actual_Vol1: -92.000000, Actual_Vol2: 333.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35488.754425, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.931033, Absolute Angle: 1.726528, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
331: 27962 [INFO] CHASSIS_PID_TURN, Time: 5068, Actual_Vol1: -92.000000, Actual_Vol2: 320.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35399.396677, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.935775, Absolute Angle: 1.726611, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
332: 27962 [INFO] CHASSIS_PID_TURN, Time: 5078, Actual_Vol1: -92.000000, Actual_Vol2: 326.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35310.014605, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938207, Absolute Angle: 1.726653, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035455, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
333: 27964 [INFO] CHASSIS_PID_TURN, Time: 5088, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: 6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35220.660700, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.935390, Absolute Angle: 1.726604, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035455, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
334: 27964 [INFO] CHASSIS_PID_TURN, Time: 5098, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35131.324033, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.933667, Absolute Angle: 1.726574, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035455, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
335: 27964 [INFO] CHASSIS_PID_TURN, Time: 5108, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35041.977290, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.934674, Absolute Angle: 1.726592, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031595, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
336: 27966 [INFO] CHASSIS_PID_TURN, Time: 5118, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34952.593748, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938354, Absolute Angle: 1.726656, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031035, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
337: 27966 [INFO] CHASSIS_PID_TURN, Time: 5128, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34863.210454, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938329, Absolute Angle: 1.726655, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031035, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
338: 27966 [INFO] CHASSIS_PID_TURN, Time: 5138, Actual_Vol1: -92.000000, Actual_Vol2: 370.000000, Actual_Vol3: 6.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34773.821228, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938923, Absolute Angle: 1.726666, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031035, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
339: 27968 [INFO] CHASSIS_PID_TURN, Time: 5148, Actual_Vol1: -92.000000, Actual_Vol2: 407.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34684.401022, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.942021, Absolute Angle: 1.726720, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028957, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
340: 27968 [INFO] CHASSIS_PID_TURN, Time: 5158, Actual_Vol1: -265.000000, Actual_Vol2: 437.000000, Actual_Vol3: -80.000000, Actual_Vol4: 74.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34594.998793, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.940223, Absolute Angle: 1.726688, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021871, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
341: 27968 [INFO] CHASSIS_PID_TURN, Time: 5168, Actual_Vol1: -136.000000, Actual_Vol2: 450.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34505.561789, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.943700, Absolute Angle: 1.726749, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019115, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
342: 27970 [INFO] CHASSIS_PID_TURN, Time: 5178, Actual_Vol1: -111.000000, Actual_Vol2: 450.000000, Actual_Vol3: -18.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34416.157872, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.940392, Absolute Angle: 1.726691, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
343: 27970 [INFO] CHASSIS_PID_TURN, Time: 5188, Actual_Vol1: -92.000000, Actual_Vol2: 450.000000, Actual_Vol3: 0.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34326.792150, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.936572, Absolute Angle: 1.726625, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
344: 27970 [INFO] CHASSIS_PID_TURN, Time: 5198, Actual_Vol1: -92.000000, Actual_Vol2: 444.000000, Actual_Vol3: 0.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34237.475382, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.931677, Absolute Angle: 1.726539, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
345: 27972 [INFO] CHASSIS_PID_TURN, Time: 5208, Actual_Vol1: -99.000000, Actual_Vol2: 456.000000, Actual_Vol3: 0.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34148.175501, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.929988, Absolute Angle: 1.726510, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
346: 27972 [INFO] CHASSIS_PID_TURN, Time: 5218, Actual_Vol1: -92.000000, Actual_Vol2: 456.000000, Actual_Vol3: 0.000000, Actual_Vol4: 99.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34058.895145, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928036, Absolute Angle: 1.726476, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
347: 27972 [INFO] CHASSIS_PID_TURN, Time: 5228, Actual_Vol1: -99.000000, Actual_Vol2: 450.000000, Actual_Vol3: 0.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33969.610486, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928466, Absolute Angle: 1.726483, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015665, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
348: 27974 [INFO] CHASSIS_PID_TURN, Time: 5238, Actual_Vol1: -86.000000, Actual_Vol2: 444.000000, Actual_Vol3: 0.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33880.362357, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.924813, Absolute Angle: 1.726419, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018887, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
349: 27974 [INFO] CHASSIS_PID_TURN, Time: 5248, Actual_Vol1: -86.000000, Actual_Vol2: 493.000000, Actual_Vol3: 6.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33791.121646, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.924071, Absolute Angle: 1.726407, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
350: 27974 [INFO] CHASSIS_PID_TURN, Time: 5258, Actual_Vol1: -283.000000, Actual_Vol2: 530.000000, Actual_Vol3: -80.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33701.849316, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.927233, Absolute Angle: 1.726462, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
351: 27976 [INFO] CHASSIS_PID_TURN, Time: 5268, Actual_Vol1: -363.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33612.548349, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.930097, Absolute Angle: 1.726512, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
352: 27976 [INFO] CHASSIS_PID_TURN, Time: 5278, Actual_Vol1: -382.000000, Actual_Vol2: 542.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33523.227118, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.932123, Absolute Angle: 1.726547, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
353: 27976 [INFO] CHASSIS_PID_TURN, Time: 5288, Actual_Vol1: -376.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33433.885374, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.934174, Absolute Angle: 1.726583, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
354: 27978 [INFO] CHASSIS_PID_TURN, Time: 5298, Actual_Vol1: -376.000000, Actual_Vol2: 548.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33344.551868, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.933351, Absolute Angle: 1.726568, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -6.800000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
355: 27978 [INFO] CHASSIS_PID_TURN, Time: 5308, Actual_Vol1: -148.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33255.254778, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.929709, Absolute Angle: 1.726505, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
356: 27978 [INFO] CHASSIS_PID_TURN, Time: 5318, Actual_Vol1: -105.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33165.977957, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.927682, Absolute Angle: 1.726470, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
357: 27980 [INFO] CHASSIS_PID_TURN, Time: 5328, Actual_Vol1: -99.000000, Actual_Vol2: 542.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33076.723304, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.925465, Absolute Angle: 1.726431, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
358: 27980 [INFO] CHASSIS_PID_TURN, Time: 5338, Actual_Vol1: -92.000000, Actual_Vol2: 585.000000, Actual_Vol3: -99.000000, Actual_Vol4: 209.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32987.518750, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.920455, Absolute Angle: 1.726343, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023245, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
359: 27980 [INFO] CHASSIS_PID_TURN, Time: 5348, Actual_Vol1: -92.000000, Actual_Vol2: 591.000000, Actual_Vol3: -92.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32898.318062, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.920069, Absolute Angle: 1.726337, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023632, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
360: 27982 [INFO] CHASSIS_PID_TURN, Time: 5358, Actual_Vol1: -86.000000, Actual_Vol2: 554.000000, Actual_Vol3: -99.000000, Actual_Vol4: 117.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32809.181388, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.913667, Absolute Angle: 1.726225, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030033, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
361: 27982 [INFO] CHASSIS_PID_TURN, Time: 5368, Actual_Vol1: -92.000000, Actual_Vol2: 542.000000, Actual_Vol3: -86.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32720.065457, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.911593, Absolute Angle: 1.726189, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028799, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
362: 27982 [INFO] CHASSIS_PID_TURN, Time: 5378, Actual_Vol1: -86.000000, Actual_Vol2: 542.000000, Actual_Vol3: -86.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32630.903980, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.916148, Absolute Angle: 1.726268, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024979, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
363: 27984 [INFO] CHASSIS_PID_TURN, Time: 5388, Actual_Vol1: -86.000000, Actual_Vol2: 542.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32541.745763, kD: 35.000000, kI: 0.0

00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.915822, Absolute Angle: 1.726263, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022581, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

364: 27984 [INFO] CHASSIS_PID_TURN, Time: 5398, Actual_Vol1: -99.000000, Actual_Vol2: 536.000000, Actual_Vol3: -160.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32452.632276, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.911349, Absolute Angle: 1.726184, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022826, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

365: 27984 [INFO] CHASSIS_PID_TURN, Time: 5408, Actual_Vol1: -290.000000, Actual_Vol2: 579.000000, Actual_Vol3: -216.000000, Actual_Vol4: 209.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32363.534246, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.909803, Absolute Angle: 1.726157, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

366: 27986 [INFO] CHASSIS_PID_TURN, Time: 5418, Actual_Vol1: -326.000000, Actual_Vol2: 634.000000, Actual_Vol3: -240.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32274.386473, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.914777, Absolute Angle: 1.726244, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, over slew: 0, Actual_Vel1: -16.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

367: 27986 [INFO] CHASSIS_PID_TURN, Time: 5428, Actual_Vol1: -370.000000, Actual_Vol2: 641.000000, Actual_Vol3: -234.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32185.245290, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.914118, Absolute Angle: 1.726233, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, over slew: 0, Actual_Vel1: -16.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

368: 27986 [INFO] CHASSIS_PID_TURN, Time: 5438, Actual_Vol1: -376.000000, Actual_Vol2: 610.000000, Actual_Vol3: -228.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32096.139350, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.910594, Absolute Angle: 1.726171, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, over slew: 0, Actual_Vel1: -7.600000, Actual_Vel2: 8.600000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

369: 27988 [INFO] CHASSIS_PID_TURN, Time: 5448, Actual_Vol1: -148.000000, Actual_Vol2: 561.000000, Actual_Vol3: -228.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32007.018162, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.912119, Absolute Angle: 1.726198, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.600000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

370: 27988 [INFO] CHASSIS_PID_TURN, Time: 5458, Actual_Vol1: -111.000000, Actual_Vol2: 548.000000, Actual_Vol3: -265.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31917.952951, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.906521, Absolute Angle: 1.726100, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027653, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.600000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

371: 27988 [INFO] CHASSIS_PID_TURN, Time: 5468, Actual_Vol1: -99.000000, Actual_Vol2: 598.000000, Actual_Vol3: -314.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31828.889803, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.906315, Absolute Angle: 1.726097, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027860, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.600000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

372: 27990 [INFO] CHASSIS_PID_TURN, Time: 5478, Actual_Vol1: -86.000000, Actual_Vol2: 628.000000, Actual_Vol3: -363.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31739.833474, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.905633, Absolute Angle: 1.726085, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028542, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000

373: 27990 [INFO] CHASSIS_PID_TURN, Time: 5488, Actual_Vol1: -86.000000, Actual_Vol2: 647.000000, Actual_Vol3: -363.000000, Actual_Vol4: 333.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31650.807550, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.902592, Absolute Angle: 1.726032, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030758, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

374: 27990 [INFO] CHASSIS_PID_TURN, Time: 5498, Actual_Vol1: -92.000000, Actual_Vol2: 647.000000, Actual_Vol3: -363.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31561.790121, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.901743, Absolute Angle: 1.726017, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027966, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

375: 27992 [INFO] CHASSIS_PID_TURN, Time: 5508, Actual_Vol1: -86.000000, Actual_Vol2: 647.000000, Actual_Vol3: -326.000000, Actual_Vol4: 333.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31472.802727, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.898739, Absolute Angle: 1.725964, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028943, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

376: 27992 [INFO] CHASSIS_PID_TURN, Time: 5518, Actual_Vol1: -86.000000, Actual_Vol2: 684.000000, Actual_Vol3: -407.000000, Actual_Vol4: 326.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31383.867080, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.893565, Absolute Angle: 1.725874, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031901, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

377: 27992 [INFO] CHASSIS_PID_TURN, Time: 5528, Actual_Vol1: -283.000000, Actual_Vol2: 752.000000, Actual_Vol3: -450.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31294.936846, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.893023, Absolute Angle: 1.725865, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027432, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

378: 27994 [INFO] CHASSIS_PID_TURN, Time: 5538, Actual_Vol1: -314.000000, Actual_Vol2: 758.000000, Actual_Vol3: -456.000000, Actual_Vol4: 450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31205.942991, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.899386, Absolute Angle: 1.725976, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027045, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -306.200000, Actual_Vel4: 0.000000

379: 27994 [INFO] CHASSIS_PID_TURN, Time: 5548, Actual_Vol1: -326.000000, Actual_Vol2: 770.000000, Actual_Vol3: -419.000000, Actual_Vol4: 462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31116.886267, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.905672, Absolute Angle: 1.726085, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023124, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -306.200000, Actual_Vel4: 0.000000

380: 27994 [INFO] CHASSIS_PID_TURN, Time: 5558, Actual_Vol1: -370.000000, Actual_Vol2: 770.000000, Actual_Vol3: -382.000000, Actual_Vol4: 480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31027.896701, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.898957, Absolute Angle: 1.725968, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023124, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -306.200000, Actual_Vel4: 28.800000

381: 27996 [INFO] CHASSIS_PID_TURN, Time: 5568, Actual_Vol1: -376.000000, Actual_Vol2: 819.000000, Actual_Vol3: -419.000000, Actual_Vol4: 419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30938.887355, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.900935, Absolute Angle: 1.726003, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023124, over slew: 0, Actual_Vel1: -17.600000, Actual_Vel2: -0.000000, Actual_Vel3: -306.200000, Actual_Vel4: 28.800000

382: 27996 [INFO] CHASSIS_PID_TURN, Time: 5578, Actual_Vol1: -456.000000, Actual_Vol2: 856.000000, Actual_Vol3: -456.000000, Actual_Vol4: 462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30849.851418, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.903594, Absolute Angle: 1.726049, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022798, over slew: 0, Actual_Vel1: -19.600000, Actual_Vel2: -0.000000, Actual_Vel3: -306.200000, Actual_Vel4: 14.800000

383: 27996 [INFO] CHASSIS_PID_TURN, Time: 5588, Actual_Vol1: -394.000000, Actual_Vol2: 862.000000, Actual_Vol3: -480.000000, Actual_Vol4: 394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30760.817773, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.903364, Absolute Angle: 1.726045, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021754, over slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -19.400000, Actual_Vel4: 13.800000

384: 27998 [INFO] CHASSIS_PID_TURN, Time: 5598, Actual_Vol1: -148.000000, Actual_Vol2: 869.000000, Actual_Vol3: -456.000000, Actual_Vol4: 277.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30671.768218, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.904956, Absolute Angle: 1.726073, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021754, over slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -19.400000, Actual_Vel4: 13.800000

385: 27998 [INFO] CHASSIS_PID_TURN, Time: 5608, Actual_Vol1: -111.000000, Actual_Vol2: 862.000000, Actual_Vol3: -456.000000, Actual_Vol4: 246.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30582.772630, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.899559, Absolute Angle: 1.725979, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021754, over slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.800000

386: 27998 [INFO] CHASSIS_PID_TURN, Time: 5618, Actual_Vol1: -92.000000, Actual_Vol2: 942.000000, Actual_Vol3: -444.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30493.780574, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.899206, Absolute Angle: 1.725973, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021095, over slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.800000

387: 28000 [INFO] CHASSIS_PID_TURN, Time: 5628, Actual_Vol1: -290.000000, Actual_Vol2: 1035.000000, Actual_Vol3: -320.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30404.784180, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.899639, Absolute Angle: 1.725980, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019095, over slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -14.200000, Actual_Vel4: 16.200000

388: 28000 [INFO] CHASSIS_PID_TURN, Time: 5638, Actual_Vol1: -314.000000, Actual_Vol2: 1066.000000, Actual_Vol3: -271.000000, Actual_Vol4: 271.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30315.813358, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.897082, Absolute Angle: 1.725935, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019095, over slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: 11.000000, Actual_Vel3: -14.200000, Actual_Vel4: 16.200000

389: 28000 [INFO] CHASSIS_PID_TURN, Time: 5648, Actual_Vol1: -370.000000, Actual_Vol2: 961.000000, Actual_Vol3: -259.000000, Actual_Vol4: 253.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30226.877258, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.893610, Absolute Angle: 1.725875, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013498, over slew: 0, Actual_Vel1: -15.000000, Actual_Vel2: 17.000000, Actual_Vel3: -14.200000, Actual_Vel4: 16.200000

390: 28002 [INFO] CHASSIS_PID_TURN, Time: 5658, Actual_Vol1: -154.000000, Actual_Vol2: 856.000000, Actual_Vol3: -283.000000, Actual_Vol4: 240.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30137.966537, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.891072, Absolute Angle: 1.725831, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015243, over slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 16.600000, Actual_Vel3: -10.800000, Actual_Vel4: 16.200000

391: 28002 [INFO] CHASSIS_PID_TURN, Time: 5668, Actual_Vol1: -99.000000, Actual_Vol2: 807.000000, Actual_Vol3: -283.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30049.050954, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.891558, Absolute Angle: 1.725839, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014600, over slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 19.200000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000

392: 28002 [INFO] CHASSIS_PID_TURN, Time: 5678, Actual_Vol1: -99.000000, Actual_Vol2: 684.000000, Actual_Vol3: -246.000000, Actual_Vol4: 246.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29960.158863, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.889209, Absolute Angle: 1.725798, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016463, over slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 19.200000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000

393: 28004 [INFO] CHASSIS_PID_TURN, Time: 5688, Actual_Vol1: -92.000000, Actual_Vol2: 641.000000, Actual_Vol3: -228.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29871.355420, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.880344, Absolute Angle: 1.725643, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025328, over slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 19.200000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000

394: 28004 [INFO] CHASSIS_PID_TURN, Time: 5698, Actual_Vol1: -209.000000, Actual_Vol2: 708.000000, Actual_Vol3: -265.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29782.571045, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.878438, Absolute Angle: 1.725610, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027235, over slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 19.200000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000

395: 28004 [INFO] CHASSIS_PID_TURN, Time: 5708, Actual_Vol1: -314.000000, Actual_Vol2: 721.000000, Actual_Vol3: -308.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29693.834033, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.873701, Absolute Angle: 1.725527, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031971, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 25.600000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000

396: 28006 [INFO] CHASSIS_PID_TURN, Time: 5718, Actual_Vol1: -363.000000, Actual_Vol2: 758.000000, Actual_Vol3: -326.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29605.189681, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.864435, Absolute Angle: 1.725366, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041237, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 25.600000, Actual_Vel3: -21.000000, Actual_Vel4: 24.600000

397: 28006 [INFO] CHASSIS_PID_TURN, Time: 5728, Actual_Vol1: -370.000000, Actual_Vol2: 764.000000, Actual_Vol3: -326.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29516.626042, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.856364, Absolute Angle: 1.725225, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.049309, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 25.600000, Actual_Vel3: -21.000000, Actual_Vel4: 50.800000

398: 28006 [INFO] CHASSIS_PID_TURN, Time: 5738, Actual_Vol1: -370.000000, Actual_Vol2: 832.000000, Actual_Vol3: -400.000000, Actual_Vol4: 333.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29428.119101, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.850694, Absolute Angle: 1.725126, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.054978, over slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 22.200000, Actual_Vel3: -21.000000, Actual_Vel4: 50.800000

399: 28008 [INFO] CHASSIS_PID_TURN, Time: 5748, Actual_Vol1: -216.000000, Actual_Vol2: 856.000000, Actual_Vol3: -437.000000, Actual_Vol4: 437.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29339.643864, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.847524, Absolute Angle: 1.725070, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 22.200000, Actual_Vel3: -43.600000, Actual_Vel4: 50.800000

400: 28008 [INFO] CHASSIS_PID_TURN, Time: 5758, Actual_Vol1: -123.000000, Actual_Vol2: 856.000000, Actual_Vol3: -400.000000, Actual_Vol4: 456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29251.127663, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.851620, Absolute Angle: 1.725142, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 22.200000, Actual_Vel3: -40.400000, Actual_Vel4: 98.400000

401: 28008 [INFO] CHASSIS_PID_TURN, Time: 5768, Actual_Vol1: -92.000000, Actual_Vol2: 856.000000, Actual_Vol3: -376.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29162.593383, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.853428, Absolute Angle: 1.725174, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 37.400000, Actual_Vel3: -40.400000, Actual_Vel4: 98.400000

402: 28010 [INFO] CHASSIS_PID_TURN, Time: 5778, Actual_Vol1: -222.000000, Actual_Vol2: 1004.000000, Actual_Vol3: -431.000000, Actual_Vol4: 462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29074.046506, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854688, Absolute Angle: 1.725196, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 37.400000, Actual_Vel3: -12.400000, Actual_Vel4: 98.400000

403: 28010 [INFO] CHASSIS_PID_TURN, Time: 5788, Actual_Vol1: -308.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -444.000000, Actual_Vol4: 499.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28985.430171, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.861633, Absolute Angle: 1.725317, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 37.400000, Actual_Vel3: -14.600000, Actual_Vel4: 98.400000

404: 28010 [INFO] CHASSIS_PID_TURN, Time: 5798, Actual_Vol1: -326.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -320.000000, Actual_Vol4: 561.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28896.820636, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.860954, Absolute Angle: 1.725305, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.052116, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: -0.000000, Actual_Vel3: -14.600000, Actual_Vel4: 98.400000

405: 28012 [INFO] CHASSIS_PID_TURN, Time: 5808, Actual_Vol1: -326.000000, Actual_Vol2: 973.000000, Actual_Vol3: -259.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28808.202645, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.861799, Absolute Angle: 1.725320, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.052116, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.600000, Actual_Vel4: 20.400000

406: 28012 [INFO] CHASSIS_PID_TURN, Time: 5818, Actual_Vol1: -413.000000, Actual_Vol2: 1029.000000, Actual_Vol3: -283.000000, Actual_Vol4: 450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28719.579633, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.862301, Absolute Angle: 1.725328, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.052116, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.600000, Actual_Vel4: 20.400000

407: 28012 [INFO] CHASSIS_PID_TURN, Time: 5828, Actual_Vol1: -450.000000, Actual_Vol2: 1053.000000, Actual_Vol3: -326.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28630.910586, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.866905, Absolute Angle: 1.725409, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.049558, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.600000, Actual_Vel4: 20.400000

408: 28014 [INFO] CHASSIS_PID_TURN, Time: 5838, Actual_Vol1: -456.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -302.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28542.318690, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.859190, Absolute Angle: 1.725274, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.046086, over slew: 0, Actual_Vel1: -16.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.000000, Actual_Vel4: 13.800000

409: 28014 [INFO] CHASSIS_PID_TURN, Time: 5848, Actual_Vol1: -413.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -302.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28453.734081, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.858461, Absolute Angle: 1.725261, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.044035, over slew: 0, Actual_Vel1: -16.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.000000, Actual_Vel4: 14.400000

410: 28014 [INFO] CHASSIS_PID_TURN, Time: 5858, Actual_Vol1: -376.000000, Actual_Vol2: 1152.000000, Actual_Vol3: -302.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28365.193451, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854063, Absolute Angle: 1.725185, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.044035, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 52.800000, Actual_Vel3: -11.600000, Actual_Vel4: 14.400000

411: 28016 [INFO] CHASSIS_PID_TURN, Time: 5868, Actual_Vol1: -370.000000, Actual_Vol2: 1177.000000, Actual_Vol3: -265.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28276.625047, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.856840, Absolute Angle: 1.725233, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041685, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 52.800000, Actual_Vel3: -11.600000, Actual_Vel4: 14.400000

412: 28016 [INFO] CHASSIS_PID_TURN, Time: 5878, Actual_Vol1: -333.000000, Actual_Vol2: 1183.000000, Actual_Vol3: -246.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28188.079172, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854588, Absolute Angle: 1.725194, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032821, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 52.800000, Actual_Vel3: -11.600000, Actual_Vel4: 14.400000

413: 28016 [INFO] CHASSIS_PID_TURN, Time: 5888, Actual_Vol1: -407.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -277.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28099.547665, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.853151, Absolute Angle: 1.725169, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030914, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 52.800000, Actual_Vel3: -11.600000, Actual_Vel4: 13.200000

414: 28018 [INFO] CHASSIS_PID_TURN, Time: 5898, Actual_Vol1: -437.000000, Actual_Vol2: 1207.000000, Actual_Vol3: -314.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28011.020204, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.852746, Absolute Angle: 1.725162, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026177, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 53.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.000000

415: 28018 [INFO] CHASSIS_PID_TURN, Time: 5908, Actual_Vol1: -450.000000, Actual_Vol2: 1195.000000, Actual_Vol3: -363.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27922.498320, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.852188, Absolute Angle: 1.725152, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019381, over slew: 0, Actual_Vel1: -12.800000, Actual_Vel2: 53.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.000000

416: 28018 [INFO] CHASSIS_PID_TURN, Time: 5918, Actual_Vol1: -450.000000, Actual_Vol2: 1090.000000, Actual_Vol3: -425.000000, Actual_Vol4: 382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27833.979691, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.851863, Absolute Angle: 1.725146, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019381, over slew: 0, Actual_Vel1: -12.800000, Actual_Vel2: 19.800000, Actual_Vel3: -11.200000, Actual_Vel4: 12.200000

417: 28020 [INFO] CHASSIS_PID_TURN, Time: 5928, Actual_Vol1: -394.000000, Actual_Vol2: 1072.000000, Actual_Vol3: -431.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27745.435924, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854377, Absolute Angle: 1.725190, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019381, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 19.800000, Actual_Vel3: -11.200000, Actual_Vel4: 12.200000

418: 28020 [INFO] CHASSIS_PID_TURN, Time: 5938, Actual_Vol1: -320.000000, Actual_Vol2: 924.000000, Actual_Vol3: -320.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27656.902057, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.853387, Absolute Angle: 1.725173, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019381, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 16.600000, Actual_Vel3: -11.800000, Actual_Vel4: 12.200000

419: 28020 [INFO] CHASSIS_PID_TURN, Time: 5948, Actual_Vol1: -308.000000, Actual_Vol2: 807.000000, Actual_Vol3: -302.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27568.343562, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.855849, Absolute Angle: 1.725216, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015285, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 16.600000, Actual_Vel3: -11.800000, Actual_Vel4: 12.200000

420: 28022 [INFO] CHASSIS_PID_TURN, Time: 5958, Actual_Vol1: -394.000000, Actual_Vol2: 838.000000, Actual_Vol3: -363.000000, Actual_Vol4: 407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27479.759651, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.858391, Absolute Angle: 1.725260, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 16.600000, Actual_Vel3: -11.800000, Actual_Vel4: 12.200000

421: 28022 [INFO] CHASSIS_PID_TURN, Time: 5968, Actual_Vol1: -431.000000, Actual_Vol2: 862.000000, Actual_Vol3: -290.000000, Actual_Vol4: 407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27391.215698, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854395, Absolute Angle: 1.725190, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 16.600000, Actual_Vel3: -17.400000, Actual_Vel4: 12.000000

422: 28022 [INFO] CHASSIS_PID_TURN, Time: 5978, Actual_Vol1: -431.000000, Actual_Vol2: 856.000000, Actual_Vol3: -246.000000, Actual_Vol4: 376.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27302.614759, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.860094, Absolute Angle: 1.725290, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 21.800000, Actual_Vel3: -17.400000, Actual_Vel4: 12.000000

423: 28024 [INFO] CHASSIS_PID_TURN, Time: 5988, Actual_Vol1: -474.000000, Actual_Vol2: 1004.000000, Actual_Vol3: -283.000000, Actual_Vol4: 413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27214.009165, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.860559, Absolute Angle: 1.725298, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 21.800000, Actual_Vel3: -17.400000, Actual_Vel4: 12.000000

424: 28024 [INFO] CHASSIS_PID_TURN, Time: 5998, Actual_Vol1: -517.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -296.000000, Actual_Vol4: 407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27125.465832, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854333, Absolute Angle: 1.725189, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 21.800000, Actual_Vel3: -22.600000, Actual_Vel4: 15.000000

425: 28024 [INFO] CHASSIS_PID_TURN, Time: 6008, Actual_Vol1: -524.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -265.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27036.943682, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.852215, Absolute Angle: 1.725152, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 18.600000, Actual_Vel3: -22.600000, Actual_Vel4: 15.000000

426: 28026 [INFO] CHASSIS_PID_TURN, Time: 6018, Actual_Vol1: -579.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -283.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26948.429185, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.851450, Absolute Angle: 1.725139, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015455, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 18.600000, Actual_Vel3: -22.600000, Actual_Vel4: 11.800000

427: 28026 [INFO] CHASSIS_PID_TURN, Time: 6028, Actual_Vol1: -628.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -320.000000, Actual_Vol4: 419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26859.965313, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.846387, Absolute Angle: 1.725051, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014172, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 18.600000, Actual_Vel3: -22.600000, Actual_Vel4: 11.800000

428: 28026 [INFO] CHASSIS_PID_TURN, Time: 6038, Actual_Vol1: -647.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -314.000000, Actual_Vol4: 302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26771.509297, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.845602, Absolute Angle: 1.725037, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014958, over slew: 0, Actual_Vel1: -450.400000, Actual_Vel2: 28.000000, Actual_Vel3: -22.600000, Actual_Vel4: 14.000000

429: 28028 [INFO] CHASSIS_PID_TURN, Time: 6048, Actual_Vol1: -598.000000, Actual_Vol2: 1244.000000, Actual_Vol3: -400.000000, Actual_Vol4: 296.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26683.064576, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.844472, Absolute Angle: 1.725017, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, over slew: 0, Actual_Vel1: -55.400000, Actual_Vel2: 28.000000, Actual_Vel3: -42.400000, Actual_Vel4: 14.000000

430: 28028 [INFO] CHASSIS_PID_TURN, Time: 6058, Actual_Vol1: -561.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -450.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26594.569134, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.849544, Absolute Angle: 1.725106, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, over slew: 0, Actual_Vel1: -55.400000, Actual_Vel2: 28.000000, Actual_Vel3: -42.400000, Actual_Vel4: 14.000000

431: 28028 [INFO] CHASSIS_PID_TURN, Time: 6068, Actual_Vol1: -517.000000, Actual_Vol2: 1281.000000, Actual_Vol3: -487.000000, Actual_Vol4: 382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26506.069906, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.849923, Absolute Angle: 1.725112, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, over slew: 0, Actual_Vel1: -21.200000, Actual_Vel2: -0.000000, Actual_Vel3: -42.400000, Actual_Vel4: 14.000000

432: 28030 [INFO] CHASSIS_PID_TURN, Time: 6078, Actual_Vol1: -474.000000, Actual_Vol2: 1349.000000, Actual_Vol3: -530.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26417.595323, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.847458, Absolute Angle: 1.725069, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 15.800000

433: 28030 [INFO] CHASSIS_PID_TURN, Time: 6088, Actual_Vol1: -462.000000, Actual_Vol2: 1368.000000, Actual_Vol3: -505.000000, Actual_Vol4: 382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26329.113469, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.848185, Absolute Angle: 1.725082, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 15.800000

434: 28030 [INFO] CHASSIS_PID_TURN, Time: 6098, Actual_Vol1: -499.000000, Actual_Vol2: 1368.000000, Actual_Vol3: -505.000000, Actual_Vol4: 283.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26240.620321, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.849315, Absolute Angle: 1.725102, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 11.800000

435: 28032 [INFO] CHASSIS_PID_TURN, Time: 6108, Actual_Vol1: -524.000000, Actual_Vol2: 1441.000000, Actual_Vol3: -536.000000, Actual_Vol4: 283.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26152.155133, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.846519, Absolute Angle: 1.725053, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 11.800000

436: 28032 [INFO] CHASSIS_PID_TURN, Time: 6118, Actual_Vol1: -530.000000, Actual_Vol2: 1460.000000, Actual_Vol3: -548.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26063.742958, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.841217, Absolute Angle: 1.724960, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019342, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 11.800000

437: 28032 [INFO] CHASSIS_PID_TURN, Time: 6128, Actual_Vol1: -536.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -604.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25975.303031, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.843993, Absolute Angle: 1.725009, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019342, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: -0.000000, Actual_Vel3: -21.400000, Actual_Vel4: 11.800000

438: 28034 [INFO] CHASSIS_PID_TURN, Time: 6138, Actual_Vol1: -542.000000, Actual_Vol2: 1614.000000, Actual_Vol3: -616.000000, Actual_Vol4: 394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25886.906303, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.839673, Absolute Angle: 1.724933, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020887, o

ver slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 31.800000, Actual_Vel3: -21.400000, Actual_Vel4: 11.800000
439: 28034 [INFO] CHASSIS_PID_TURN, Time: 6148, Actual_Vol1: -499.000000, Actual_Vol2: 1435.000000, Actual_Vol3: -542.000000, Actual_Vol4: 450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25798.502533, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.840377, Absolute Angle: 1.724946, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020887, o ver slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 27.400000, Actual_Vel3: -17.800000, Actual_Vel4: 11.800000
440: 28034 [INFO] CHASSIS_PID_TURN, Time: 6158, Actual_Vol1: -511.000000, Actual_Vol2: 1411.000000, Actual_Vol3: -487.000000, Actual_Vol4: 493.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25710.116081, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.838645, Absolute Angle: 1.724916, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021914, o ver slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 31.400000, Actual_Vel3: -17.800000, Actual_Vel4: 14.200000
441: 28036 [INFO] CHASSIS_PID_TURN, Time: 6168, Actual_Vol1: -530.000000, Actual_Vol2: 1331.000000, Actual_Vol3: -480.000000, Actual_Vol4: 542.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25621.753442, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.836264, Absolute Angle: 1.724874, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024296, o ver slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 29.800000, Actual_Vel3: -17.800000, Actual_Vel4: 14.200000
442: 28036 [INFO] CHASSIS_PID_TURN, Time: 6178, Actual_Vol1: -585.000000, Actual_Vol2: 1164.000000, Actual_Vol3: -505.000000, Actual_Vol4: 554.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25533.447218, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.830622, Absolute Angle: 1.724776, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.029937, o ver slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 29.800000, Actual_Vel3: -17.800000, Actual_Vel4: 10.800000
443: 28036 [INFO] CHASSIS_PID_TURN, Time: 6188, Actual_Vol1: -622.000000, Actual_Vol2: 1177.000000, Actual_Vol3: -505.000000, Actual_Vol4: 567.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25445.140879, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.830634, Absolute Angle: 1.724776, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023711, o ver slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 21.600000, Actual_Vel3: -10.800000, Actual_Vel4: 10.200000
444: 28038 [INFO] CHASSIS_PID_TURN, Time: 6198, Actual_Vol1: -616.000000, Actual_Vol2: 1072.000000, Actual_Vol3: -456.000000, Actual_Vol4: 524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25356.865868, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.827501, Absolute Angle: 1.724721, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024714, o ver slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 21.600000, Actual_Vel3: -10.800000, Actual_Vel4: 10.200000
445: 28038 [INFO] CHASSIS_PID_TURN, Time: 6208, Actual_Vol1: -665.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -474.000000, Actual_Vol4: 487.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25268.601636, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.826423, Absolute Angle: 1.724702, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025026, o ver slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 21.600000, Actual_Vel3: -10.800000, Actual_Vel4: 11.000000
446: 28038 [INFO] CHASSIS_PID_TURN, Time: 6218, Actual_Vol1: -653.000000, Actual_Vol2: 1177.000000, Actual_Vol3: -517.000000, Actual_Vol4: 456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25180.345854, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.825578, Absolute Angle: 1.724687, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024345, o ver slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 18.000000, Actual_Vel3: -10.800000, Actual_Vel4: 12.600000
447: 28040 [INFO] CHASSIS_PID_TURN, Time: 6228, Actual_Vol1: -665.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -567.000000, Actual_Vol4: 394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25092.212236, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.813362, Absolute Angle: 1.724474, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036561, o ver slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 18.000000, Actual_Vel3: -10.600000, Actual_Vel4: 12.600000
448: 28040 [INFO] CHASSIS_PID_TURN, Time: 6238, Actual_Vol1: -708.000000, Actual_Vol2: 1244.000000, Actual_Vol3: -616.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25004.088807, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.812343, Absolute Angle: 1.724456, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.037580, o ver slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 18.000000, Actual_Vel3: -10.600000, Actual_Vel4: 13.200000
449: 28040 [INFO] CHASSIS_PID_TURN, Time: 6248, Actual_Vol1: -702.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -622.000000, Actual_Vol4: 413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24916.008324, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.808048, Absolute Angle: 1.724382, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041875, o ver slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 16.800000, Actual_Vel3: -10.600000, Actual_Vel4: 13.200000
450: 28042 [INFO] CHASSIS_PID_TURN, Time: 6258, Actual_Vol1: -838.000000, Actual_Vol2: 1349.000000, Actual_Vol3: -659.000000, Actual_Vol4: 419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24827.967721, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.804060, Absolute Angle: 1.724312, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.045862, o ver slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 16.800000, Actual_Vel3: -9.600000, Actual_Vel4: 11.400000
451: 28042 [INFO] CHASSIS_PID_TURN, Time: 6268, Actual_Vol1: -912.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -708.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24739.990815, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.797691, Absolute Angle: 1.724201, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051624, o ver slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 16.800000, Actual_Vel3: -9.600000, Actual_Vel4: 11.400000
452: 28042 [INFO] CHASSIS_PID_TURN, Time: 6278, Actual_Vol1: -936.000000, Actual_Vol2: 1361.000000, Actual_Vol3: -708.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24652.064836, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.792598, Absolute Angle: 1.724112, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.056717, o ver slew: 0, Actual_Vel1: -8.000000, Actual_Vel2: 10.200000, Actual_Vel3: -8.400000, Actual_Vel4: 11.400000
453: 28044 [INFO] CHASSIS_PID_TURN, Time: 6288, Actual_Vol1: -936.000000, Actual_Vol2: 1435.000000, Actual_Vol3: -721.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24564.204620, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.786022, Absolute Angle: 1.723997, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.063293, o ver slew: 0, Actual_Vel1: -8.000000, Actual_Vel2: 10.200000, Actual_Vel3: -8.400000, Actual_Vel4: 9.600000
454: 28044 [INFO] CHASSIS_PID_TURN, Time: 6298, Actual_Vol1: -936.000000, Actual_Vol2: 1460.000000, Actual_Vol3: -708.000000, Actual_Vol4: 413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24476.378882, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.782574, Absolute Angle: 1.723937, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.063945, o ver slew: 0, Actual_Vel1: -8.000000, Actual_Vel2: 10.200000, Actual_Vel3: -8.400000, Actual_Vel4: 9.600000
455: 28044 [INFO] CHASSIS_PID_TURN, Time: 6308, Actual_Vol1: -936.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -788.000000, Actual_Vol4: 407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24388.542597, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.783629, Absolute Angle: 1.723955, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.061419, o ver slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: -0.000000, Actual_Vel3: -8.400000, Actual_Vel4: 9.600000
456: 28046 [INFO] CHASSIS_PID_TURN, Time: 6318, Actual_Vol1: -832.000000, Actual_Vol2: 1595.000000, Actual_Vol3: -875.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24300.697033, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.784556, Absolute Angle: 1.723972, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.061419, o ver slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -11.200000, Actual_Vel4: 9.600000
457: 28046 [INFO] CHASSIS_PID_TURN, Time: 6328, Actual_Vol1: -862.000000, Actual_Vol2: 1602.000000, Actual_Vol3: -967.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24212.869739, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.782729, Absolute Angle: 1.723940, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057803, o ver slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -11.200000, Actual_Vel4: 9.600000
458: 28046 [INFO] CHASSIS_PID_TURN, Time: 6338, Actual_Vol1: -912.000000, Actual_Vol2: 1675.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 493.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24125.041576, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.782816, Absolute Angle: 1.723941, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057803, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -11.200000, Actual_Vel4: 10.600000
459: 28048 [INFO] CHASSIS_PID_TURN, Time: 6348, Actual_Vol1: -924.000000, Actual_Vol2: 1694.000000, Actual_Vol3: -992.000000, Actual_Vol4: 511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24037.247026, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.779455, Absolute Angle: 1.723882, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.059190, o ver slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -8.800000, Actual_Vel4: 10.600000
460: 28048 [INFO] CHASSIS_PID_TURN, Time: 6358, Actual_Vol1: -986.000000, Actual_Vol2: 1768.000000, Actual_Vol3: -992.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23949.478992, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.776803, Absolute Angle: 1.723836, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.059460, o ver slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -8.800000, Actual_Vel4: 10.600000
461: 28048 [INFO] CHASSIS_PID_TURN, Time: 6368, Actual_Vol1: -1029.000000, Actual_Vol2: 1793.000000, Actual_Vol3: -998.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23861.754896, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.772410, Absolute Angle: 1.723760, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.058224, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.600000, Actual_Vel4: 9.800000
462: 28050 [INFO] CHASSIS_PID_TURN, Time: 6378, Actual_Vol1: -1078.000000, Actual_Vol2: 1786.000000, Actual_Vol3: -986.000000, Actual_Vol4: 474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23774.139047, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.761585, Absolute Angle: 1.723571, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.069049, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.600000, Actual_Vel4: 12.600000
463: 28050 [INFO] CHASSIS_PID_TURN, Time: 6388, Actual_Vol1: -1066.000000, Actual_Vol2: 1866.000000, Actual_Vol3: -1035.000000, Actual_Vol4: 456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23686.636837, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.750221, Absolute Angle: 1.723372, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.077280 , over slew: 0, Actual_Vel1: -21.800000, Actual_Vel2: -0.000000, Actual_Vel3: -9.600000, Actual_Vel4: 12.600000
464: 28050 [INFO] CHASSIS_PID_TURN, Time: 6398, Actual_Vol1: -986.000000, Actual_Vol2: 1885.000000, Actual_Vol3: -1072.000000, Actual_Vol4: 468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23599.252516, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.738432, Absolute Angle: 1.723166, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.087991, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.600000, Actual_Vel4: 12.600000
465: 28052 [INFO] CHASSIS_PID_TURN, Time: 6408, Actual_Vol1: -986.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23511.961261, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.729126, Absolute Angle: 1.723004, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.096453, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.000000, Actual_Vel4: 10.400000
466: 28052 [INFO] CHASSIS_PID_TURN, Time: 6418, Actual_Vol1: -1016.000000, Actual_Vol2: 2014.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23424.708154, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.725311, Absolute Angle: 1.722937, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.088051 , over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.000000, Actual_Vel4: 9.800000
467: 28052 [INFO] CHASSIS_PID_TURN, Time: 6428, Actual_Vol1: -1072.000000, Actual_Vol2: 2168.000000, Actual_Vol3: -1201.000000, Actual_Vol4: 474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23337.549230, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.715892, Absolute Angle: 1.722773, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.096451 , over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 8.000000, Actual_Vel3: -9.000000, Actual_Vel4: 9.800000
468: 28054 [INFO] CHASSIS_PID_TURN, Time: 6438, Actual_Vol1: -1146.000000, Actual_Vol2: 2076.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23250.513279, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.703595, Absolute Angle: 1.722558, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.104453 , over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 15.400000, Actual_Vel3: -9.000000, Actual_Vel4: 9.800000
469: 28054 [INFO] CHASSIS_PID_TURN, Time: 6448, Actual_Vol1: -1201.000000, Actual_Vol2: 1977.000000, Actual_Vol3: -1300.000000, Actual_Vol4: 493.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23163.594978, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.691830, Absolute Angle: 1.722353, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.112230 , over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 15.400000, Actual_Vel3: -9.000000, Actual_Vel4: 9.800000
470: 28054 [INFO] CHASSIS_PID_TURN, Time: 6458, Actual_Vol1: -1244.000000, Actual_Vol2: 1947.000000, Actual_Vol3: -1349.000000, Actual_Vol4: 536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23076.780670, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.681431, Absolute Angle: 1.722172, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.116260 , over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 15.400000, Actual_Vel3: -9.000000, Actual_Vel4: 12.000000
471: 28056 [INFO] CHASSIS_PID_TURN, Time: 6468, Actual_Vol1: -1300.000000, Actual_Vol2: 2008.000000, Actual_Vol3: -1386.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22990.091853, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.668882, Absolute Angle: 1.721953, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.123716 , over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 15.400000, Actual_Vel3: -258.200000, Actual_Vel4: 12.000000
472: 28056 [INFO] CHASSIS_PID_TURN, Time: 6478, Actual_Vol1: -1331.000000, Actual_Vol2: 2020.000000, Actual_Vol3: -1398.000000, Actual_Vol4: 480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22903.544889, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.654696, Absolute Angle: 1.721705, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.131325 , over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 16.200000, Actual_Vel3: -258.200000, Actual_Vel4: 16.400000
473: 28056 [INFO] CHASSIS_PID_TURN, Time: 6488, Actual_Vol1: -1392.000000, Actual_Vol2: 1947.000000, Actual_Vol3: -1368.000000, Actual_Vol4: 462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22817.177327, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.636756, Absolute Angle: 1.721392, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.147800 , over slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: 16.200000, Actual_Vel3: -24.600000, Actual_Vel4: 16.400000
474: 28058 [INFO] CHASSIS_PID_TURN, Time: 6498, Actual_Vol1: -1435.000000, Actual_Vol2: 1996.000000, Actual_Vol3: -1374.000000, Actual_Vol4: 499.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22731.025856, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.615147, Absolute Angle: 1.721015, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.169409 , over slew: 0, Actual_Vel1: -21.800000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.400000
475: 28058 [INFO] CHASSIS_PID_TURN, Time: 6508, Actual_Vol1: -1349.000000, Actual_Vol2: 1940.000000, Actual_Vol3: -1368.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22645.041083, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.598477, Absolute Angle: 1.720724, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.186079 , over slew: 0, Actual_Vel1: -27.600000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.400000
476: 28058 [INFO] CHASSIS_PID_TURN, Time: 6518, Actual_Vol1: -1324.000000, Actual_Vol2: 1996.000000, Actual_Vol3: -1411.000000, Actual_Vol4: 579.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22560.175881, kD: 35.000000

, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.486520, Absolute Angle: 1.718770, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.296296, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.800000

477: 28060 [INFO] CHASSIS_PID_TURN, Time: 6528, Actual_Vol1: -1294.000000, Actual_Vol2: 2014.000000, Actual_Vol3: -1454.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22475.498954, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.467693, Absolute Angle: 1.718441, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.315124, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.800000

478: 28060 [INFO] CHASSIS_PID_TURN, Time: 6538, Actual_Vol1: -1318.000000, Actual_Vol2: 2144.000000, Actual_Vol3: -1552.000000, Actual_Vol4: 671.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22390.979511, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.451944, Absolute Angle: 1.718166, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.327511, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.800000

479: 28060 [INFO] CHASSIS_PID_TURN, Time: 6548, Actual_Vol1: -1337.000000, Actual_Vol2: 2193.000000, Actual_Vol3: -1583.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22306.596291, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.438322, Absolute Angle: 1.717929, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.338482, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 15.600000, Actual_Vel3: -24.600000, Actual_Vel4: 19.200000

480: 28062 [INFO] CHASSIS_PID_TURN, Time: 6558, Actual_Vol1: -1417.000000, Actual_Vol2: 2359.000000, Actual_Vol3: -1669.000000, Actual_Vol4: 678.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22222.315306, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.428099, Absolute Angle: 1.717750, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.344311, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 15.600000, Actual_Vel3: -24.600000, Actual_Vel4: 19.200000

481: 28062 [INFO] CHASSIS_PID_TURN, Time: 6568, Actual_Vol1: -1429.000000, Actual_Vol2: 2396.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22138.147880, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.416743, Absolute Angle: 1.717552, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.344842, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 15.600000, Actual_Vel3: -16.000000, Actual_Vel4: 15.400000

482: 28062 [INFO] CHASSIS_PID_TURN, Time: 6578, Actual_Vol1: -1361.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22054.139384, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.400850, Absolute Angle: 1.717275, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.349371, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 11.600000, Actual_Vel3: -16.000000, Actual_Vel4: 15.400000

483: 28064 [INFO] CHASSIS_PID_TURN, Time: 6588, Actual_Vol1: -1417.000000, Actual_Vol2: 2458.000000, Actual_Vol3: -1786.000000, Actual_Vol4: 721.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21970.283071, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.385631, Absolute Angle: 1.717009, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.352801, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 10.400000, Actual_Vel3: -16.000000, Actual_Vel4: 15.400000

484: 28064 [INFO] CHASSIS_PID_TURN, Time: 6598, Actual_Vol1: -1435.000000, Actual_Vol2: 2470.000000, Actual_Vol3: -1712.000000, Actual_Vol4: 825.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21886.548662, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.373441, Absolute Angle: 1.716796, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.355685, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 10.400000, Actual_Vel3: -10.600000, Actual_Vel4: 15.400000

485: 28064 [INFO] CHASSIS_PID_TURN, Time: 6608, Actual_Vol1: -1558.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -1706.000000, Actual_Vol4: 881.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21803.005613, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.354305, Absolute Angle: 1.716462, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.371006, over slew: 0, Actual_Vel1: -13.000000, Actual_Vel2: 11.200000, Actual_Vel3: -10.600000, Actual_Vel4: 15.400000

486: 28066 [INFO] CHASSIS_PID_TURN, Time: 6618, Actual_Vol1: -1571.000000, Actual_Vol2: 2519.000000, Actual_Vol3: -1688.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21719.668685, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.333693, Absolute Angle: 1.716102, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.382200, over slew: 0, Actual_Vel1: -13.000000, Actual_Vel2: 11.200000, Actual_Vel3: -11.400000, Actual_Vel4: 15.400000

487: 28066 [INFO] CHASSIS_PID_TURN, Time: 6628, Actual_Vol1: -1651.000000, Actual_Vol2: 2526.000000, Actual_Vol3: -1688.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21636.589191, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.307949, Absolute Angle: 1.715653, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.395646, over slew: 0, Actual_Vel1: -13.000000, Actual_Vel2: 11.200000, Actual_Vel3: -11.400000, Actual_Vel4: 15.400000

488: 28066 [INFO] CHASSIS_PID_TURN, Time: 6638, Actual_Vol1: -1669.000000, Actual_Vol2: 2519.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 979.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21553.728296, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.286090, Absolute Angle: 1.715272, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.405741, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 11.200000, Actual_Vel3: -11.400000, Actual_Vel4: 14.800000

489: 28068 [INFO] CHASSIS_PID_TURN, Time: 6648, Actual_Vol1: -1731.000000, Actual_Vol2: 2526.000000, Actual_Vol3: -1817.000000, Actual_Vol4: 986.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21471.152349, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.257595, Absolute Angle: 1.714774, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.423836, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 11.200000, Actual_Vel3: -11.400000, Actual_Vel4: 14.800000

490: 28068 [INFO] CHASSIS_PID_TURN, Time: 6658, Actual_Vol1: -1749.000000, Actual_Vol2: 2526.000000, Actual_Vol3: -1817.000000, Actual_Vol4: 986.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21388.770019, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.238233, Absolute Angle: 1.714436, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.430649, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 11.200000, Actual_Vel3: -11.600000, Actual_Vel4: 14.800000

491: 28068 [INFO] CHASSIS_PID_TURN, Time: 6668, Actual_Vol1: -1756.000000, Actual_Vol2: 2599.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21306.580032, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.218999, Absolute Angle: 1.714101, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.435698, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: 11.200000, Actual_Vel3: -11.600000, Actual_Vel4: 14.800000

492: 28070 [INFO] CHASSIS_PID_TURN, Time: 6678, Actual_Vol1: -1817.000000, Actual_Vol2: 2655.000000, Actual_Vol3: -1823.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21224.569612, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.201042, Absolute Angle: 1.713787, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.435714, over slew: 0, Actual_Vel1: -15.400000, Actual_Vel2: 11.200000, Actual_Vel3: -11.600000, Actual_Vel4: 28.400000

493: 28070 [INFO] CHASSIS_PID_TURN, Time: 6688, Actual_Vol1: -1793.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1041.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21142.721531, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.184808, Absolute Angle: 1.713504, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.430339, over slew: 0, Actual_Vel1: -15.400000, Actual_Vel2: 10.600000, Actual_Vel3: -11.600000, Actual_Vel4: 28.400000

494: 28070 [INFO] CHASSIS_PID_TURN, Time: 6698, Actual_Vol1: -1768.000000, Actual_Vol2: 2636.000000, Actual_Vol3: -1953.000000, Actual_Vol4: 1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21061.113085, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.160845, Absolute Angle: 1.713086, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.437633, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 11.600000, Actual_Vel3: -11.600000, Actual_Vel4: 28.400000

495: 28072 [INFO] CHASSIS_PID_TURN, Time: 6708, Actual_Vol1: -1768.000000, Actual_Vol2: 2618.000000, Actual_Vol3: -1953.000000, Actual_Vol4: 1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20979.799318, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.131377, Absolute Angle: 1.712571, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.355143, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 11.600000, Actual_Vel3: -15.400000, Actual_Vel4: 28.400000

496: 28072 [INFO] CHASSIS_PID_TURN, Time: 6718, Actual_Vol1: -1848.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -1959.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20898.764056, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.103526, Absolute Angle: 1.712085, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.364167, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 11.600000, Actual_Vel3: -15.400000, Actual_Vel4: 28.400000

497: 28072 [INFO] CHASSIS_PID_TURN, Time: 6728, Actual_Vol1: -1860.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20818.037827, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.072623, Absolute Angle: 1.711546, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.379321, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 13.800000, Actual_Vel3: -15.400000, Actual_Vel4: 20.200000

498: 28074 [INFO] CHASSIS_PID_TURN, Time: 6738, Actual_Vol1: -1873.000000, Actual_Vol2: 2766.000000, Actual_Vol3: -2014.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20737.683655, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.035417, Absolute Angle: 1.710897, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.402905, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 13.800000, Actual_Vel3: -16.600000, Actual_Vel4: 20.200000

499: 28074 [INFO] CHASSIS_PID_TURN, Time: 6748, Actual_Vol1: -1965.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1201.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20657.687055, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.999660, Absolute Angle: 1.710272, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.428439, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: 13.800000, Actual_Vel3: -17.200000, Actual_Vel4: 20.200000

500: 28074 [INFO] CHASSIS_PID_TURN, Time: 6758, Actual_Vol1: -1873.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1244.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20578.007844, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.967921, Absolute Angle: 1.709719, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.448821, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: 13.800000, Actual_Vel3: -17.200000, Actual_Vel4: 21.200000

501: 28076 [INFO] CHASSIS_PID_TURN, Time: 6768, Actual_Vol1: -1965.000000, Actual_Vol2: 2809.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1244.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20498.584374, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.942347, Absolute Angle: 1.709272, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.458503, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: 18.600000, Actual_Vel3: -17.200000, Actual_Vel4: 21.200000

502: 28076 [INFO] CHASSIS_PID_TURN, Time: 6778, Actual_Vol1: -1990.000000, Actual_Vol2: 2809.000000, Actual_Vol3: -2014.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20419.341475, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.924290, Absolute Angle: 1.708957, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.461341, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: 18.600000, Actual_Vel3: -17.200000, Actual_Vel4: 21.200000

503: 28076 [INFO] CHASSIS_PID_TURN, Time: 6788, Actual_Vol1: -2131.000000, Actual_Vol2: 2809.000000, Actual_Vol3: -2119.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20340.304506, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.903697, Absolute Angle: 1.708598, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.469744, over slew: 0, Actual_Vel1: -9.600000, Actual_Vel2: 18.600000, Actual_Vel3: -17.200000, Actual_Vel4: 21.200000

504: 28078 [INFO] CHASSIS_PID_TURN, Time: 6798, Actual_Vol1: -2094.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2137.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20261.400795, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.890371, Absolute Angle: 1.708365, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.463934, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.600000, Actual_Vel3: -13.600000, Actual_Vel4: 20.200000

505: 28078 [INFO] CHASSIS_PID_TURN, Time: 6808, Actual_Vol1: -2008.000000, Actual_Vol2: 2834.000000, Actual_Vol3: -2020.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20182.675766, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.872503, Absolute Angle: 1.708053, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.461190, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.600000, Actual_Vel3: -13.600000, Actual_Vel4: 20.200000

506: 28078 [INFO] CHASSIS_PID_TURN, Time: 6818, Actual_Vol1: -2205.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -2119.000000, Actual_Vol4: 1287.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20104.264479, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.841129, Absolute Angle: 1.707506, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.466821, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.600000, Actual_Vel3: -13.600000, Actual_Vel4: 20.200000

507: 28080 [INFO] CHASSIS_PID_TURN, Time: 6828, Actual_Vol1: -2236.000000, Actual_Vol2: 2852.000000, Actual_Vol3: -2261.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20026.145173, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.811931, Absolute Angle: 1.706996, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.474159, over slew: 0, Actual_Vel1: -16.000000, Actual_Vel2: 9.600000, Actual_Vel3: -20.200000, Actual_Vel4: 20.200000

508: 28080 [INFO] CHASSIS_PID_TURN, Time: 6838, Actual_Vol1: -2255.000000, Actual_Vol2: 2914.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19948.392846, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.775233, Absolute Angle: 1.706355, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.482362, over slew: 0, Actual_Vel1: -16.000000, Actual_Vel2: 9.600000, Actual_Vel3: -20.200000, Actual_Vel4: 27.800000

509: 28080 [INFO] CHASSIS_PID_TURN, Time: 6848, Actual_Vol1: -2248.000000, Actual_Vol2: 2871.000000, Actual_Vol3: -2285.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19871.019388, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.737346, Absolute Angle: 1.705694, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.500887, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.000000, Actual_Vel3: -20.200000, Actual_Vel4: 27.800000

510: 28082 [INFO] CHASSIS_PID_TURN, Time: 6858, Actual_Vol1: -2119.000000, Actual_Vol2: 2864.000000, Actual_Vol3: -2347.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19794.025191, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.699420, Absolute Angle: 1.705032, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.519579, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.000000, Actual_Vel3: -20.200000, Actual_Vel4: 27.800000

511: 28082 [INFO] CHASSIS_PID_TURN, Time: 6868, Actual_Vol1: -2236.000000, Actual_Vol2: 2901.000000, Actual_Vol3: -2359.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19717.425872, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.659932, Absolute Angle: 1.704343, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.541110, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.000000, Actual_Vel3: -44.600000, Actual_Vel4: 27.800000

512: 28082 [INFO] CHASSIS_PID_TURN, Time: 6878, Actual_Vol1: -2273.000000, Actual_Vol2: 2907.000000, Actual_Vol3: -2168.000000, Actual_Vol4: 1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19641.175841, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.625003, Absolute Angle: 1.703733, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.559805, over slew: 0, Actual_Vel1: -20.400000, Actual_Vel2: 12.000000, Actual_Vel3: -44.600000, Actual_Vel4: 24.200000

513: 28084 [INFO] CHASSIS_PID_TURN, Time: 6888, Actual_Vol1: -2353.000000, Actual_Vol2: 2907.000000, Actual_Vol3: -2156.000000, Actual_Vol4: 1398.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19565.258232, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.591761, Absolute Angle: 1.703153, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.569084, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 12.000000, Actual_Vel3: -44.600000, Actual_Vel4: 24.200000

514: 28084 [INFO] CHASSIS_PID_TURN, Time: 6898, Actual_Vol1: -2113.000000, Actual_Vol2: 2907.000000, Actual_Vol3: -2248.000000, Actual_Vol4: 1349.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19489.750380, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.550785, Absolute Angle: 1.702438, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.580591, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 12.000000, Actual_Vel3: -44.600000, Actual_Vel4: 21.000000

515: 28084 [INFO] CHASSIS_PID_TURN, Time: 6908, Actual_Vol1: -2094.000000, Actual_Vol2: 3000.000000, Actual_Vol3: -2279.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19414.628522, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.512186, Absolute Angle: 1.701764, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.591340, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 12.000000, Actual_Vel3: -36.200000, Actual_Vel4: 21.000000

516: 28086 [INFO] CHASSIS_PID_TURN, Time: 6918, Actual_Vol1: -2211.000000, Actual_Vol2: 3135.000000, Actual_Vol3: -2156.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19339.893009, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.473551, Absolute Angle: 1.701090, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.599072, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 12.000000, Actual_Vel3: -36.200000, Actual_Vel4: 21.000000

517: 28086 [INFO] CHASSIS_PID_TURN, Time: 6928, Actual_Vol1: -2255.000000, Actual_Vol2: 3172.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1423.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19265.584254, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.430875, Absolute Angle: 1.700345, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.604542, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 15.000000, Actual_Vel3: -36.200000, Actual_Vel4: 21.000000

518: 28086 [INFO] CHASSIS_PID_TURN, Time: 6938, Actual_Vol1: -2353.000000, Actual_Vol2: 3092.000000, Actual_Vol3: -2211.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19191.800149, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.378411, Absolute Angle: 1.699430, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.621249, over slew: 0, Actual_Vel1: -42.400000, Actual_Vel2: 20.800000, Actual_Vel3: -17.000000, Actual_Vel4: 27.400000

519: 28088 [INFO] CHASSIS_PID_TURN, Time: 6948, Actual_Vol1: -2255.000000, Actual_Vol2: 2932.000000, Actual_Vol3: -2051.000000, Actual_Vol4: 1349.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19118.534870, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.326528, Absolute Angle: 1.698524, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.641393, over slew: 0, Actual_Vel1: -29.800000, Actual_Vel2: 20.800000, Actual_Vel3: -15.400000, Actual_Vel4: 23.000000

520: 28088 [INFO] CHASSIS_PID_TURN, Time: 6958, Actual_Vol1: -2107.000000, Actual_Vol2: 2994.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1349.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19045.788150, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.274672, Absolute Angle: 1.697619, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.666675, over slew: 0, Actual_Vel1: -29.800000, Actual_Vel2: 20.800000, Actual_Vel3: -15.400000, Actual_Vel4: 23.000000

521: 28088 [INFO] CHASSIS_PID_TURN, Time: 6968, Actual_Vol1: -2230.000000, Actual_Vol2: 3154.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18973.535126, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.225302, Absolute Angle: 1.696757, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.698988, over slew: 0, Actual_Vel1: -29.800000, Actual_Vel2: 20.800000, Actual_Vel3: -15.400000, Actual_Vel4: 23.000000

522: 28090 [INFO] CHASSIS_PID_TURN, Time: 6978, Actual_Vol1: -2255.000000, Actual_Vol2: 3105.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1392.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18901.800732, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.173439, Absolute Angle: 1.695852, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.730258, over slew: 0, Actual_Vel1: -29.800000, Actual_Vel2: 22.800000, Actual_Vel3: -10.400000, Actual_Vel4: 23.000000

523: 28090 [INFO] CHASSIS_PID_TURN, Time: 6988, Actual_Vol1: -2267.000000, Actual_Vol2: 2938.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18830.648053, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.115268, Absolute Angle: 1.694837, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.775103, over slew: 0, Actual_Vel1: -27.000000, Actual_Vel2: 22.800000, Actual_Vel3: -10.400000, Actual_Vel4: 23.800000

524: 28090 [INFO] CHASSIS_PID_TURN, Time: 6998, Actual_Vol1: -2255.000000, Actual_Vol2: 2920.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1404.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18760.018225, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.062983, Absolute Angle: 1.693924, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.809520, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 22.800000, Actual_Vel3: -10.400000, Actual_Vel4: 23.800000

525: 28092 [INFO] CHASSIS_PID_TURN, Time: 7008, Actual_Vol1: -2107.000000, Actual_Vol2: 2932.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1355.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18689.854057, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.016417, Absolute Angle: 1.693112, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.824712, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 22.800000, Actual_Vel3: -8.800000, Actual_Vel4: 18.600000

526: 28092 [INFO] CHASSIS_PID_TURN, Time: 7018, Actual_Vol1: -2082.000000, Actual_Vol2: 3000.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1349.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18620.222499, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.963156, Absolute Angle: 1.692182, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.848775, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 22.800000, Actual_Vel3: -11.400000, Actual_Vel4: 18.600000

527: 28092 [INFO] CHASSIS_PID_TURN, Time: 7028, Actual_Vol1: -2230.000000, Actual_Vol2: 3160.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18551.055688, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.916681, Absolute Angle: 1.691371, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.858552, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 23.200000, Actual_Vel3: -11.400000, Actual_Vel4: 12.400000

528: 28094 [INFO] CHASSIS_PID_TURN, Time: 7038, Actual_Vol1: -2248.000000, Actual_Vol2: 3043.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18482.350623, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.870507, Absolute Angle: 1.690565, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.866839, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 28.800000, Actual_Vel3: -9.800000, Actual_Vel4: 12.400000

529: 28094 [INFO] CHASSIS_PID_TURN, Time: 7048, Actual_Vol1: -2236.000000, Actual_Vol2: 2901.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18414.125482, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.822514, Absolute Angle: 1.689727, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.876906, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 28.800000, Actual_Vel3: -13.000000, Actual_Vel4: 10.800000

530: 28094 [INFO] CHASSIS_PID_TURN, Time: 7058, Actual_Vol1: -2347.000000, Actual_Vol2: 2907.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 1306.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18346.385799, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.773968, Absolute Angle: 1.688880, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.885964, over slew: 0, Actual_Vel1: -31.800000, Actual_Vel2: 28.800000, Actual_Vel3: -13.000000, Actual_Vel4: 10.800000

531: 28096 [INFO] CHASSIS_PID_TURN, Time: 7068, Actual_Vol1: -2255.000000, Actual_Vol2: 2864.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18279.163702, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.722210, Absolute Angle: 1.687977, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.902793, over slew: 0, Actual_Vel1: -30.000000, Actual_Vel2: 42.200000, Actual_Vel3: -13.000000, Actual_Vel4: 10.800000

532: 28096 [INFO] CHASSIS_PID_TURN, Time: 7078, Actual_Vol1: -2119.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18212.465678, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.669802, Absolute Angle: 1.687062, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.921959, over slew: 0, Actual_Vel1: -30.000000, Actual_Vel2: 31.800000, Actual_Vel3: -13.000000, Actual_Vel4: 9.400000

533: 28096 [INFO] CHASSIS_PID_TURN, Time: 7088, Actual_Vol1: -2230.000000, Actual_Vol2: 2827.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1263.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18146.374480, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.609120, Absolute Angle: 1.686003, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.941665, over slew: 0, Actual_Vel1: -30.000000, Actual_Vel2: 31.800000, Actual_Vel3: -15.400000, Actual_Vel4: 9.400000

534: 28098 [INFO] CHASSIS_PID_TURN, Time: 7098, Actual_Vol1: -2261.000000, Actual_Vol2: 2827.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1207.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18080.880736, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.549374, Absolute Angle: 1.684960, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.962811, over slew: 0, Actual_Vel1: -31.800000, Actual_Vel2: 31.800000, Actual_Vel3: -16.200000, Actual_Vel4: 8.800000

535: 28098 [INFO] CHASSIS_PID_TURN, Time: 7108, Actual_Vol1: -2242.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -1940.000000, Actual_Vol4: 1214.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18015.977599, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.490314, Absolute Angle: 1.683929, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.983238, over slew: 0, Actual_Vel1: -31.800000, Actual_Vel2: 38.400000, Actual_Vel3: -16.200000, Actual_Vel4: 8.800000

536: 28098 [INFO] CHASSIS_PID_TURN, Time: 7118, Actual_Vol1: -2113.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1885.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17951.665703, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.431190, Absolute Angle: 1.682898, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.999686, over slew: 0, Actual_Vel1: -31.800000, Actual_Vel2: 38.400000, Actual_Vel3: -16.200000, Actual_Vel4: 8.800000

537: 28100 [INFO] CHASSIS_PID_TURN, Time: 7128, Actual_Vol1: -2230.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 1220.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17887.961687, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.370402, Absolute Angle: 1.681837, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.008009, over slew: 0, Actual_Vel1: -19.000000, Actual_Vel2: 38.400000, Actual_Vel3: -16.200000, Actual_Vel4: 11.600000

538: 28100 [INFO] CHASSIS_PID_TURN, Time: 7138, Actual_Vol1: -2113.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1207.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17824.790376, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.317131, Absolute Angle: 1.680907, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.009397, over slew: 0, Actual_Vel1: -19.000000, Actual_Vel2: 21.600000, Actual_Vel3: -21.600000, Actual_Vel4: 11.600000

539: 28100 [INFO] CHASSIS_PID_TURN, Time: 7148, Actual_Vol1: -2082.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1934.000000, Actual_Vol4: 1201.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17762.115028, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.267535, Absolute Angle: 1.680041, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.007137, over slew: 0, Actual_Vel1: -19.000000, Actual_Vel2: 21.600000, Actual_Vel3: -22.000000, Actual_Vel4: 12.400000

540: 28102 [INFO] CHASSIS_PID_TURN, Time: 7158, Actual_Vol1: -2082.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1805.000000, Actual_Vol4: 1158.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17699.898180, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.221685, Absolute Angle: 1.679241, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.003618, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 14.200000, Actual_Vel3: -22.000000, Actual_Vel4: 12.400000

541: 28102 [INFO] CHASSIS_PID_TURN, Time: 7168, Actual_Vol1: -2070.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17638.145174, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.175301, Absolute Angle: 1.678431, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.998139, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 14.200000, Actual_Vel3: -22.000000, Actual_Vel4: 12.400000

542: 28102 [INFO] CHASSIS_PID_TURN, Time: 7178, Actual_Vol1: -2076.000000, Actual_Vol2: 2827.000000, Actual_Vol3: -1879.000000, Actual_Vol4: 1158.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17576.895138, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.125004, Absolute Angle: 1.677554, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.990264, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 10.800000, Actual_Vel3: -22.000000, Actual_Vel4: 19.000000

543: 28104 [INFO] CHASSIS_PID_TURN, Time: 7188, Actual_Vol1: -2057.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -1879.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17516.165089, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.073005, Absolute Angle: 1.676646, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.989978, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 10.800000, Actual_Vel3: -22.000000, Actual_Vel4: 23.200000

544: 28104 [INFO] CHASSIS_PID_TURN, Time: 7198, Actual_Vol1: -1922.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17455.895975, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.026911, Absolute Angle: 1.675842, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.989505, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 10.800000, Actual_Vel3: -22.000000, Actual_Vel4: 23.200000

545: 28104 [INFO] CHASSIS_PID_TURN, Time: 7208, Actual_Vol1: -1971.000000, Actual_Vol2: 2760.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 1041.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17396.082127, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.981385, Absolute Angle: 1.675047, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.981771, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 8.000000, Actual_Vel3: -22.000000, Actual_Vel4: 23.200000

546: 28106 [INFO] CHASSIS_PID_TURN, Time: 7218, Actual_Vol1: -2057.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17336.642565, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.943956, Absolute Angle: 1.674394, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.972725, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 9.000000, Actual_Vel3: -22.000000, Actual_Vel4: 23.200000

547: 28106 [INFO] CHASSIS_PID_TURN, Time: 7228, Actual_Vol1: -1996.000000, Actual_Vol2: 2630.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17277.602454, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.904011, Absolute Angle: 1.673697, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.966495, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 9.000000, Actual_Vel3: -22.000000, Actual_Vel4: 19.800000

548: 28106 [INFO] CHASSIS_PID_TURN, Time: 7238, Actual_Vol1: -1990.000000, Actual_Vol2: 2544.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17218.940171, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.866228, Absolute Angle: 1.673037, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.956286, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.000000, Actual_Vel3: -16.200000, Actual_Vel4: 19.800000

549: 28108 [INFO] CHASSIS_PID_TURN, Time: 7248, Actual_Vol1: -2002.000000, Actual_Vol2: 2532.000000, Actual_Vol3: -1793.000000, Actual_Vol4: 992.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17160.582078, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.835809, Absolute Angle: 1.672506, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.938159, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.000000, Actual_Vel3: -16.200000, Actual_Vel4: 19.400000

550: 28108 [INFO] CHASSIS_PID_TURN, Time: 7258, Actual_Vol1: -2205.000000, Actual_Vol2: 2599.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17102.514432, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.806765, Absolute Angle: 1.671999, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.915445, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 9.000000, Actual_Vel3: -16.200000, Actual_Vel4: 19.400000

551: 28108 [INFO] CHASSIS_PID_TURN, Time: 7268, Actual_Vol1: -2088.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17044.654985, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.785945, Absolute Angle: 1.671636, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.883858

, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.800000, Actual_Vel3: -16.200000, Actual_Vel4: 19.400000
552: 28110 [INFO] CHASSIS_PID_TURN, Time: 7278, Actual_Vol1: -1879.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 986.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16987.074088, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.758090, Absolute Angle: 1.671150, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.851030, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.800000, Actual_Vel3: -18.200000, Actual_Vel4: 19.400000
553: 28110 [INFO] CHASSIS_PID_TURN, Time: 7288, Actual_Vol1: -1866.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 1016.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16929.739933, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.733416, Absolute Angle: 1.670719, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.815959, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.800000, Actual_Vel3: -18.200000, Actual_Vel4: 19.400000
554: 28110 [INFO] CHASSIS_PID_TURN, Time: 7298, Actual_Vol1: -1860.000000, Actual_Vol2: 2741.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16872.628174, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.711176, Absolute Angle: 1.670331, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.779138, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 24.600000, Actual_Vel3: -18.200000, Actual_Vel4: 14.800000
555: 28112 [INFO] CHASSIS_PID_TURN, Time: 7308, Actual_Vol1: -1971.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 979.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16815.792085, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.683609, Absolute Angle: 1.669850, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.747581, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 24.600000, Actual_Vel3: -18.200000, Actual_Vel4: 17.800000
556: 28112 [INFO] CHASSIS_PID_TURN, Time: 7318, Actual_Vol1: -2051.000000, Actual_Vol2: 2747.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16759.239316, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.655277, Absolute Angle: 1.669355, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.715125, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 24.600000, Actual_Vel3: -18.200000, Actual_Vel4: 17.800000
557: 28112 [INFO] CHASSIS_PID_TURN, Time: 7328, Actual_Vol1: -2070.000000, Actual_Vol2: 2741.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 973.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16702.943962, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.629535, Absolute Angle: 1.668906, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.687596, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 54.200000, Actual_Vel3: -18.200000, Actual_Vel4: 17.800000
558: 28114 [INFO] CHASSIS_PID_TURN, Time: 7338, Actual_Vol1: -2076.000000, Actual_Vol2: 2618.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 973.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16646.846913, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.609705, Absolute Angle: 1.668560, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.657830, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 54.200000, Actual_Vel3: -31.600000, Actual_Vel4: 21.800000
559: 28114 [INFO] CHASSIS_PID_TURN, Time: 7348, Actual_Vol1: -2082.000000, Actual_Vol2: 2532.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16591.008960, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.583795, Absolute Angle: 1.668108, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.637890, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 54.200000, Actual_Vel3: -31.600000, Actual_Vel4: 21.800000
560: 28114 [INFO] CHASSIS_PID_TURN, Time: 7358, Actual_Vol1: -2224.000000, Actual_Vol2: 2593.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16535.441837, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.556712, Absolute Angle: 1.667635, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.618588, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 54.200000, Actual_Vel3: -31.600000, Actual_Vel4: 21.800000
561: 28116 [INFO] CHASSIS_PID_TURN, Time: 7368, Actual_Vol1: -2242.000000, Actual_Vol2: 2667.000000, Actual_Vol3: -1842.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16480.104536, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.533730, Absolute Angle: 1.667234, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.591274, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -14.200000, Actual_Vel4: 21.800000
562: 28116 [INFO] CHASSIS_PID_TURN, Time: 7378, Actual_Vol1: -1959.000000, Actual_Vol2: 2717.000000, Actual_Vol3: -1793.000000, Actual_Vol4: 973.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16425.034990, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.506955, Absolute Angle: 1.666767, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.566050, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -14.200000, Actual_Vel4: 21.800000
563: 28116 [INFO] CHASSIS_PID_TURN, Time: 7388, Actual_Vol1: -1866.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16370.254565, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.478043, Absolute Angle: 1.666262, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.548869, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -14.200000, Actual_Vel4: 43.600000
564: 28118 [INFO] CHASSIS_PID_TURN, Time: 7398, Actual_Vol1: -1860.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 986.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16315.804386, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.445018, Absolute Angle: 1.665686, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.536367, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -14.200000, Actual_Vel4: 33.000000
565: 28118 [INFO] CHASSIS_PID_TURN, Time: 7408, Actual_Vol1: -1959.000000, Actual_Vol2: 2760.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16261.657866, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.414652, Absolute Angle: 1.665156, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.529304, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -15.200000, Actual_Vel4: 33.000000
566: 28118 [INFO] CHASSIS_PID_TURN, Time: 7418, Actual_Vol1: -1971.000000, Actual_Vol2: 2803.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16207.772419, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.388545, Absolute Angle: 1.664700, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.515466, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -15.200000, Actual_Vel4: 33.000000
567: 28120 [INFO] CHASSIS_PID_TURN, Time: 7428, Actual_Vol1: -2045.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 967.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16154.081454, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.369096, Absolute Angle: 1.664360, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.497132, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 32.200000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
568: 28120 [INFO] CHASSIS_PID_TURN, Time: 7438, Actual_Vol1: -2064.000000, Actual_Vol2: 2624.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 1004.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16100.624707, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.345675, Absolute Angle: 1.663952, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.490134, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 19.400000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
569: 28120 [INFO] CHASSIS_PID_TURN, Time: 7448, Actual_Vol1: -2070.000000, Actual_Vol2: 2526.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16047.419153, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.320555, Absolute Angle: 1.663513, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.486209, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 19.400000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
570: 28122 [INFO] CHASSIS_PID_TURN, Time: 7458, Actual_Vol1: -2230.000000, Actual_Vol2: 2612.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15994.363039, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.305611, Absolute Angle: 1.663252, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.480333, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 19.400000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
571: 28122 [INFO] CHASSIS_PID_TURN, Time: 7468, Actual_Vol1: -2193.000000, Actual_Vol2: 2704.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15941.535097, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.282794, Absolute Angle: 1.662854, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.475295, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 19.400000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
572: 28122 [INFO] CHASSIS_PID_TURN, Time: 7478, Actual_Vol1: -2101.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15888.930142, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.260496, Absolute Angle: 1.662465, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.472920, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 19.400000, Actual_Vel3: -10.800000, Actual_Vel4: 22.800000
573: 28124 [INFO] CHASSIS_PID_TURN, Time: 7488, Actual_Vol1: -2088.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15836.518122, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.241202, Absolute Angle: 1.662128, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.469974, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 19.400000, Actual_Vel3: -10.800000, Actual_Vel4: 19.200000
574: 28124 [INFO] CHASSIS_PID_TURN, Time: 7498, Actual_Vol1: -2020.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15784.340827, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.217729, Absolute Angle: 1.661719, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.465879, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 19.400000, Actual_Vel3: -10.800000, Actual_Vel4: 19.200000
575: 28124 [INFO] CHASSIS_PID_TURN, Time: 7508, Actual_Vol1: -1971.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15732.392410, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.194842, Absolute Angle: 1.661319, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.460435, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 19.400000, Actual_Vel3: -10.800000, Actual_Vel4: 19.200000
576: 28126 [INFO] CHASSIS_PID_TURN, Time: 7518, Actual_Vol1: -2020.000000, Actual_Vol2: 2784.000000, Actual_Vol3: -1632.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15680.758196, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.163421, Absolute Angle: 1.660771, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.466114, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 19.400000, Actual_Vel3: -19.200000, Actual_Vel4: 26.800000
577: 28126 [INFO] CHASSIS_PID_TURN, Time: 7528, Actual_Vol1: -1977.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1602.000000, Actual_Vol4: 850.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15629.415303, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.134289, Absolute Angle: 1.660262, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.475416, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 11.000000, Actual_Vel3: -19.200000, Actual_Vel4: 26.800000
578: 28126 [INFO] CHASSIS_PID_TURN, Time: 7538, Actual_Vol1: -2002.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -1743.000000, Actual_Vol4: 838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15578.360528, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.105478, Absolute Angle: 1.659759, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.478318, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 11.000000, Actual_Vel3: -19.200000, Actual_Vel4: 26.800000
579: 28128 [INFO] CHASSIS_PID_TURN, Time: 7548, Actual_Vol1: -2057.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15527.519889, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.084064, Absolute Angle: 1.659386, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.472649, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 11.000000, Actual_Vel3: -14.000000, Actual_Vel4: 26.800000
580: 28128 [INFO] CHASSIS_PID_TURN, Time: 7558, Actual_Vol1: -2070.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 912.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15476.916654, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.060324, Absolute Angle: 1.658971, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.473406, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 7.400000, Actual_Vel3: -14.000000, Actual_Vel4: 26.800000
581: 28128 [INFO] CHASSIS_PID_TURN, Time: 7568, Actual_Vol1: -2131.000000, Actual_Vol2: 2741.000000, Actual_Vol3: -1786.000000, Actual_Vol4: 918.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15426.495463, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.042119, Absolute Angle: 1.658654, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.464836, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 7.400000, Actual_Vel3: -14.000000, Actual_Vel4: 43.400000
582: 28130 [INFO] CHASSIS_PID_TURN, Time: 7578, Actual_Vol1: -2230.000000, Actual_Vol2: 2784.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15376.238192, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.025727, Absolute Angle: 1.658368, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.452315, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 7.400000, Actual_Vel3: -14.000000, Actual_Vel4: 43.400000
583: 28130 [INFO] CHASSIS_PID_TURN, Time: 7588, Actual_Vol1: -2255.000000, Actual_Vol2: 2760.000000, Actual_Vol3: -1786.000000, Actual_Vol4: 825.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15326.107341, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.013085, Absolute Angle: 1.658147, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.431933, over slew: 0, Actual_Vel1: -7.600000, Actual_Vel2: 8.000000, Actual_Vel3: -14.000000, Actual_Vel4: 43.400000
584: 28130 [INFO] CHASSIS_PID_TURN, Time: 7598, Actual_Vol1: -2248.000000, Actual_Vol2: 2741.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 832.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15276.052654, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.005469, Absolute Angle: 1.658014, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.409183, over slew: 0, Actual_Vel1: -7.600000, Actual_Vel2: 8.000000, Actual_Vel3: -14.000000, Actual_Vel4: 43.400000
585: 28132 [INFO] CHASSIS_PID_TURN, Time: 7608, Actual_Vol1: -2261.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 832.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15226.086710, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.996594, Absolute Angle: 1.657859, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.391950, over slew: 0, Actual_Vel1: -7.600000, Actual_Vel2: 8.000000, Actual_Vel3: -35.000000, Actual_Vel4: 43.400000
586: 28132 [INFO] CHASSIS_PID_TURN, Time: 7618, Actual_Vol1: -2255.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15176.222639, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.986407, Absolute Angle: 1.657681, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.382689, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 8.000000, Actual_Vel3: -35.000000, Actual_Vel4: 23.800000
587: 28132 [INFO] CHASSIS_PID_TURN, Time: 7628, Actual_Vol1: -2255.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 912.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15126.466296, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.975634, Absolute Angle: 1.657493, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.370041, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 8.000000, Actual_Vel3: -35.000000, Actual_Vel4: 23.800000
588: 28134 [INFO] CHASSIS_PID_TURN, Time: 7638, Actual_Vol1: -2310.000000, Actual_Vol2: 2575.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15076.897263, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.956903, Absolute Angle: 1.657166, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.363652, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 11.000000, Actual_Vel3: -0.000000, Actual_Vel4: 23.800000
589: 28134 [INFO] CHASSIS_PID_TURN, Time: 7648, Actual_Vol1: -2359.000000, Actual_Vol2: 2624.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15027.518531, kD: 35.000000

, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.937873, Absolute Angle: 1.656834, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.367738 , over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 11.000000, Actual_Vel3: -0.000000, Actual_Vel4: 23.800000

590: 28134 [INFO] CHASSIS_PID_TURN, Time: 7658, Actual_Vol1: -2365.000000, Actual_Vol2: 2698.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 832.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14978.297407, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.922112, Absolute Angle: 1.656559, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.360682 , over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 11.000000, Actual_Vel3: -0.000000, Actual_Vel4: 15.800000

591: 28136 [INFO] CHASSIS_PID_TURN, Time: 7668, Actual_Vol1: -2365.000000, Actual_Vol2: 2575.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14929.189940, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.910747, Absolute Angle: 1.656361, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.349749 , over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 9.000000, Actual_Vel3: -33.600000, Actual_Vel4: 15.800000

592: 28136 [INFO] CHASSIS_PID_TURN, Time: 7678, Actual_Vol1: -2341.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -1928.000000, Actual_Vol4: 850.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14881.243700, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.794624, Absolute Angle: 1.654334, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.446578 , over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 9.000000, Actual_Vel3: -33.600000, Actual_Vel4: 15.800000

593: 28136 [INFO] CHASSIS_PID_TURN, Time: 7688, Actual_Vol1: -2267.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 918.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14833.455263, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.778844, Absolute Angle: 1.654059, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.438886 , over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 9.000000, Actual_Vel3: -33.600000, Actual_Vel4: 15.800000

594: 28138 [INFO] CHASSIS_PID_TURN, Time: 7698, Actual_Vol1: -610.000000, Actual_Vol2: 782.000000, Actual_Vol3: -425.000000, Actual_Vol4: 179.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14785.806834, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.764843, Absolute Angle: 1.653814, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.429999, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 4.200000, Actual_Vel3: -33.600000, Actual_Vel4: 15.800000

595: 28138 [INFO] CHASSIS_PID_TURN, Time: 7708, Actual_Vol1: -136.000000, Actual_Vol2: 197.000000, Actual_Vol3: -80.000000, Actual_Vol4: 37.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14738.271980, kD: 35.000000, kI: 0. 000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.753485, Absolute Angle: 1.653616, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.409936, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 15.800000

596: 28138 [INFO] CHASSIS_PID_TURN, Time: 7718, Actual_Vol1: -92.000000, Actual_Vol2: 296.000000, Actual_Vol3: -18.000000, Actual_Vol4: 12.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14690.784542, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.748744, Absolute Angle: 1.653533, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.385545, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.000000

597: 28140 [INFO] CHASSIS_PID_TURN, Time: 7728, Actual_Vol1: -277.000000, Actual_Vol2: 431.000000, Actual_Vol3: -80.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14643.245777, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.753877, Absolute Angle: 1.653623, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.356734, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 6.800000, Actual_Vel4: 8.000000

598: 28140 [INFO] CHASSIS_PID_TURN, Time: 7738, Actual_Vol1: -308.000000, Actual_Vol2: 450.000000, Actual_Vol3: -92.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14595.546183, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.769959, Absolute Angle: 1.653904, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.335320, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -4.200000, Actual_Vel3: 6.800000, Actual_Vel4: 8.000000

599: 28140 [INFO] CHASSIS_PID_TURN, Time: 7748, Actual_Vol1: -314.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14547.612216, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.793397, Absolute Angle: 1.654313, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.311580, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -4.200000, Actual_Vel3: 6.800000, Actual_Vel4: 0.000000

600: 28142 [INFO] CHASSIS_PID_TURN, Time: 7758, Actual_Vol1: -320.000000, Actual_Vol2: 450.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14499.526499, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.808572, Absolute Angle: 1.654577, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.293375, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -4.200000, Actual_Vel3: 6.800000, Actual_Vel4: 0.000000

601: 28142 [INFO] CHASSIS_PID_TURN, Time: 7768, Actual_Vol1: -320.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14451.318271, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.820823, Absolute Angle: 1.654791, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.276983, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -24.400000, Actual_Vel3: 6.800000, Actual_Vel4: 0.000000

602: 28142 [INFO] CHASSIS_PID_TURN, Time: 7778, Actual_Vol1: -320.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14403.092581, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.822569, Absolute Angle: 1.654822, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.264341, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -24.400000, Actual_Vel3: 6.800000, Actual_Vel4: 0.000000

603: 28144 [INFO] CHASSIS_PID_TURN, Time: 7788, Actual_Vol1: -314.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14354.863668, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.822891, Absolute Angle: 1.654827, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.256725, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -24.400000, Actual_Vel3: 17.200000, Actual_Vel4: 0.000000

604: 28144 [INFO] CHASSIS_PID_TURN, Time: 7798, Actual_Vol1: -320.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14306.645176, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.821849, Absolute Angle: 1.654809, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.247851, over s lew: 0, Actual_Vel1: 8.800000, Actual_Vel2: -17.000000, Actual_Vel3: 17.200000, Actual_Vel4: 0.000000

605: 28144 [INFO] CHASSIS_PID_TURN, Time: 7808, Actual_Vol1: -314.000000, Actual_Vol2: 456.000000, Actual_Vol3: -86.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14258.452662, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.819251, Absolute Angle: 1.654764, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.237663, over s lew: 0, Actual_Vel1: 8.800000, Actual_Vel2: -17.000000, Actual_Vel3: 17.200000, Actual_Vel4: 0.000000

606: 28146 [INFO] CHASSIS_PID_TURN, Time: 7818, Actual_Vol1: -314.000000, Actual_Vol2: 493.000000, Actual_Vol3: -92.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14210.350980, kD: 35.000000, kI: 0. 000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.810168, Absolute Angle: 1.654605, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.226890, over slew: 0, Actual_Vel1: 8.800000, Actual_Vel2: -17.000000, Actual_Vel3: 9.800000, Actual_Vel4: 0.000000

607: 28146 [INFO] CHASSIS_PID_TURN, Time: 7828, Actual_Vol1: -413.000000, Actual_Vol2: 536.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14162.229456, kD: 35.000000, kI: 0. 000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.812152, Absolute Angle: 1.654640, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.208160, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 9.800000, Actual_Vel4: 0.000000

608: 28146 [INFO] CHASSIS_PID_TURN, Time: 7838, Actual_Vol1: -431.000000, Actual_Vol2: 536.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14114.110500, kD: 35.000000, kI: 0. 000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.811896, Absolute Angle: 1.654636, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.189129, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 9.800000, Actual_Vel4: 0.000000

609: 28148 [INFO] CHASSIS_PID_TURN, Time: 7848, Actual_Vol1: -437.000000, Actual_Vol2: 536.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14065.923561, kD: 35.000000, kI: 0. 000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.818694, Absolute Angle: 1.654754, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.173369, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

610: 28148 [INFO] CHASSIS_PID_TURN, Time: 7858, Actual_Vol1: -431.000000, Actual_Vol2: 542.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14017.770319, kD: 35.000000, kI: 0. 000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.815324, Absolute Angle: 1.654695, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.162003, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

611: 28148 [INFO] CHASSIS_PID_TURN, Time: 7868, Actual_Vol1: -474.000000, Actual_Vol2: 585.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13969.580198, kD: 35.000000, kI: 0. 000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.819012, Absolute Angle: 1.654760, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

612: 28150 [INFO] CHASSIS_PID_TURN, Time: 7878, Actual_Vol1: -499.000000, Actual_Vol2: 628.000000, Actual_Vol3: -191.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13921.390719, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.818948, Absolute Angle: 1.654759, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

613: 28150 [INFO] CHASSIS_PID_TURN, Time: 7888, Actual_Vol1: -517.000000, Actual_Vol2: 641.000000, Actual_Vol3: -222.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13873.232923, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.815780, Absolute Angle: 1.654703, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

614: 28150 [INFO] CHASSIS_PID_TURN, Time: 7898, Actual_Vol1: -511.000000, Actual_Vol2: 634.000000, Actual_Vol3: -228.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13825.090933, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.814199, Absolute Angle: 1.654676, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

615: 28152 [INFO] CHASSIS_PID_TURN, Time: 7908, Actual_Vol1: -511.000000, Actual_Vol2: 634.000000, Actual_Vol3: -222.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13776.958291, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.813264, Absolute Angle: 1.654659, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

616: 28152 [INFO] CHASSIS_PID_TURN, Time: 7918, Actual_Vol1: -554.000000, Actual_Vol2: 678.000000, Actual_Vol3: -222.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13728.866303, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.809199, Absolute Angle: 1.654588, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.069015, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

617: 28152 [INFO] CHASSIS_PID_TURN, Time: 7928, Actual_Vol1: -610.000000, Actual_Vol2: 715.000000, Actual_Vol3: -283.000000, Actual_Vol4: 197.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13680.792486, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807382, Absolute Angle: 1.654557, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.052932, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

618: 28154 [INFO] CHASSIS_PID_TURN, Time: 7938, Actual_Vol1: -616.000000, Actual_Vol2: 752.000000, Actual_Vol3: -302.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13632.751935, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.804055, Absolute Angle: 1.654499, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.029495, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

619: 28154 [INFO] CHASSIS_PID_TURN, Time: 7948, Actual_Vol1: -616.000000, Actual_Vol2: 758.000000, Actual_Vol3: -308.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13584.681295, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807064, Absolute Angle: 1.654551, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018836, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

620: 28154 [INFO] CHASSIS_PID_TURN, Time: 7958, Actual_Vol1: -616.000000, Actual_Vol2: 752.000000, Actual_Vol3: -308.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13536.540629, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.814067, Absolute Angle: 1.654673, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018836, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

621: 28156 [INFO] CHASSIS_PID_TURN, Time: 7968, Actual_Vol1: -653.000000, Actual_Vol2: 838.000000, Actual_Vol3: -308.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13488.373965, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.816666, Absolute Angle: 1.654719, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018836, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

622: 28156 [INFO] CHASSIS_PID_TURN, Time: 7978, Actual_Vol1: -690.000000, Actual_Vol2: 924.000000, Actual_Vol3: -407.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13440.230935, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.814303, Absolute Angle: 1.654678, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018836, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

623: 28156 [INFO] CHASSIS_PID_TURN, Time: 7988, Actual_Vol1: -702.000000, Actual_Vol2: 942.000000, Actual_Vol3: -425.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13392.073555, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.815738, Absolute Angle: 1.654703, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017794, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

624: 28158 [INFO] CHASSIS_PID_TURN, Time: 7998, Actual_Vol1: -702.000000, Actual_Vol2: 942.000000, Actual_Vol3: -431.000000, Actual_Vol4: 326.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13343.880530, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.819303, Absolute Angle: 1.654765, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

625: 28158 [INFO] CHASSIS_PID_TURN, Time: 8008, Actual_Vol1: -702.000000, Actual_Vol2: 936.000000, Actual_Vol3: -431.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13295.756307, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.812422, Absolute Angle: 1.654645, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

626: 28158 [INFO] CHASSIS_PID_TURN, Time: 8018, Actual_Vol1: -819.000000, Actual_Vol2: 992.000000, Actual_Vol3: -431.000000, Actual_Vol4: 333.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13247.690020, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.806629, Absolute Angle: 1.654544, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

627: 28160 [INFO] CHASSIS_PID_TURN, Time: 8028, Actual_Vol1: -912.000000, Actual_Vol2: 1035.000000, Actual_Vol3: -505.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13199.646593, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.804343, Absolute Angle: 1.654504, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

628: 28160 [INFO] CHASSIS_PID_TURN, Time: 8038, Actual_Vol1: -924.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -517.000000, Actual_Vol4: 450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13151.568288, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807830, Absolute Angle: 1.654565, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

629: 28160 [INFO] CHASSIS_PID_TURN, Time: 8048, Actual_Vol1: -924.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -517.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13103.490114, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807817, Absolute Angle: 1.654564, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

630: 28162 [INFO] CHASSIS_PID_TURN, Time: 8058, Actual_Vol1: -967.000000, Actual_Vol2: 1090.000000, Actual_Vol3: -517.000000, Actual_Vol4: 456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13055.409430, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.808068, Absolute Angle: 1.654569, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

631: 28162 [INFO] CHASSIS_PID_TURN, Time: 8068, Actual_Vol1: -1010.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -598.000000, Actual_Vol4: 524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13007.267906, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.814152, Absolute Angle: 1.654675, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

632: 28162 [INFO] CHASSIS_PID_TURN, Time: 8078, Actual_Vol1: -1029.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -616.000000, Actual_Vol4: 536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12959.153057, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.811485, Absolute Angle: 1.654628, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

633: 28164 [INFO] CHASSIS_PID_TURN, Time: 8088, Actual_Vol1: -1016.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -610.000000, Actual_Vol4: 542.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12911.038301, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.811476, Absolute Angle: 1.654628, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

634: 28164 [INFO] CHASSIS_PID_TURN, Time: 8098, Actual_Vol1: -1023.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -610.000000, Actual_Vol4: 536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12862.882038, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.815626, Absolute Angle: 1.654701, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

635: 28164 [INFO] CHASSIS_PID_TURN, Time: 8108, Actual_Vol1: -1060.000000, Actual_Vol2: 1214.000000, Actual_Vol3: -610.000000, Actual_Vol4: 616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12814.804135, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807790, Absolute Angle: 1.654564, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

636: 28166 [INFO] CHASSIS_PID_TURN, Time: 8118, Actual_Vol1: -1103.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -696.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12766.771454, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.803268, Absolute Angle: 1.654485, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016034, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

637: 28166 [INFO] CHASSIS_PID_TURN, Time: 8128, Actual_Vol1: -1152.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -702.000000, Actual_Vol4: 628.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12718.777104, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.799435, Absolute Angle: 1.654418, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

638: 28166 [INFO] CHASSIS_PID_TURN, Time: 8138, Actual_Vol1: -1152.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -708.000000, Actual_Vol4: 628.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12670.782526, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.799458, Absolute Angle: 1.654418, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

639: 28168 [INFO] CHASSIS_PID_TURN, Time: 8148, Actual_Vol1: -1183.000000, Actual_Vol2: 1312.000000, Actual_Vol3: -702.000000, Actual_Vol4: 628.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12622.743310, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.803922, Absolute Angle: 1.654496, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

640: 28168 [INFO] CHASSIS_PID_TURN, Time: 8158, Actual_Vol1: -1238.000000, Actual_Vol2: 1343.000000, Actual_Vol3: -893.000000, Actual_Vol4: 696.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12574.691970, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.805134, Absolute Angle: 1.654517, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

641: 28168 [INFO] CHASSIS_PID_TURN, Time: 8168, Actual_Vol1: -1238.000000, Actual_Vol2: 1324.000000, Actual_Vol3: -764.000000, Actual_Vol4: 715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12526.648875, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.804309, Absolute Angle: 1.654503, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

642: 28170 [INFO] CHASSIS_PID_TURN, Time: 8178, Actual_Vol1: -1250.000000, Actual_Vol2: 1275.000000, Actual_Vol3: -708.000000, Actual_Vol4: 721.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12478.575820, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807306, Absolute Angle: 1.654555, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

643: 28170 [INFO] CHASSIS_PID_TURN, Time: 8188, Actual_Vol1: -1287.000000, Actual_Vol2: 1306.000000, Actual_Vol3: -702.000000, Actual_Vol4: 715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12430.496531, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807929, Absolute Angle: 1.654566, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

644: 28170 [INFO] CHASSIS_PID_TURN, Time: 8198, Actual_Vol1: -1324.000000, Actual_Vol2: 1343.000000, Actual_Vol3: -899.000000, Actual_Vol4: 906.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12382.387943, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.810859, Absolute Angle: 1.654617, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

645: 28172 [INFO] CHASSIS_PID_TURN, Time: 8208, Actual_Vol1: -1337.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -930.000000, Actual_Vol4: 918.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12334.288670, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.809927, Absolute Angle: 1.654601, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

646: 28172 [INFO] CHASSIS_PID_TURN, Time: 8218, Actual_Vol1: -1337.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -930.000000, Actual_Vol4: 924.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12286.207463, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.808121, Absolute Angle: 1.654570, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

647: 28172 [INFO] CHASSIS_PID_TURN, Time: 8228, Actual_Vol1: -1374.000000, Actual_Vol2: 1398.000000, Actual_Vol3: -936.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12238.120661, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.808680, Absolute Angle: 1.654579, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

648: 28174 [INFO] CHASSIS_PID_TURN, Time: 8238, Actual_Vol1: -1423.000000, Actual_Vol2: 1448.000000, Actual_Vol3: -1016.000000, Actual_Vol4: 1010.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12190.047165, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807350, Absolute Angle: 1.654556, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

649: 28174 [INFO] CHASSIS_PID_TURN, Time: 8248, Actual_Vol1: -1429.000000, Actual_Vol2: 1448.000000, Actual_Vol3: -1035.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12142.019942, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.802722, Absolute Angle: 1.654475, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000

650: 28174 [INFO] CHASSIS_PID_TURN, Time: 8258, Actual_Vol1: -1441.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1041.000000, Actual_Vol4: 961.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12094.056572, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.796337, Absolute Angle: 1.654364, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019289, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 7.600000

651: 28176 [INFO] CHASSIS_PID_TURN, Time: 8268, Actual_Vol1: -1466.000000, Actual_Vol2: 1534.000000, Actual_Vol3: -1035.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12046.129698, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.792687, Absolute Angle: 1.654300, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022939, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 7.600000

652: 28176 [INFO] CHASSIS_PID_TURN, Time: 8278, Actual_Vol1: -1558.000000, Actual_Vol2: 1583.000000, Actual_Vol3: -1146.000000, Actual_Vol4: 1016.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11998.159657, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.797004, Absolute Angle: 1.654376, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022939, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 7.600000

653: 28176 [INFO] CHASSIS_PID_TURN, Time: 8288, Actual_Vol1: -1565.000000, Actual_Vol2: 1595.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 1041.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11950.202759, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.795690, Absolute Angle: 1.654353, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022939, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 7.600000

654: 28178 [INFO] CHASSIS_PID_TURN, Time: 8298, Actual_Vol1: -1565.000000, Actual_Vol2: 1595.000000, Actual_Vol3: -1164.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11902.273446, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.792931, Absolute Angle: 1.654305, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018171, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000

655: 28178 [INFO] CHASSIS_PID_TURN, Time: 8308, Actual_Vol1: -1602.000000, Actual_Vol2: 1639.000000, Actual_Vol3: -1164.000000, Actual_Vol4: 1041.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11854.335486, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.793796, Absolute Angle: 1.654320, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018171, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000

656: 28178 [INFO] CHASSIS_PID_TURN, Time: 8318, Actual_Vol1: -1657.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1238.000000, Actual_Vol4: 1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11806.405436, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.793005, Absolute Angle: 1.654306, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018171, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000

657: 28180 [INFO] CHASSIS_PID_TURN, Time: 8328, Actual_Vol1: -1663.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1263.000000, Actual_Vol4: 1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11758.527678, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.787776, Absolute Angle: 1.654215, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023083, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000

658: 28180 [INFO] CHASSIS_PID_TURN, Time: 8338, Actual_Vol1: -1669.000000, Actual_Vol2: 1694.000000, Actual_Vol3: -1263.000000, Actual_Vol4: 1158.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11710.621645, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.790603, Absolute Angle: 1.654264, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023083, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000

659: 28180 [INFO] CHASSIS_PID_TURN, Time: 8348, Actual_Vol1: -1706.000000, Actual_Vol2: 1731.000000, Actual_Vol3: -1263.000000, Actual_Vol4: 1238.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11662.681530, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.794012, Absolute Angle: 1.654323, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023083, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000

660: 28182 [INFO] CHASSIS_PID_TURN, Time: 8358, Actual_Vol1: -1749.000000, Actual_Vol2: 1774.000000, Actual_Vol3: -1343.000000, Actual_Vol4: 1263.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11614.792331, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.788920, Absolute Angle: 1.654234, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023083, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000

661: 28182 [INFO] CHASSIS_PID_TURN, Time: 8368, Actual_Vol1: -1762.000000, Actual_Vol2: 1780.000000, Actual_Vol3: -1355.000000, Actual_Vol4: 1269.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11566.939926, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.785241, Absolute Angle: 1.654170, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025618, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000

662: 28182 [INFO] CHASSIS_PID_TURN, Time: 8378, Actual_Vol1: -1799.000000, Actual_Vol2: 1823.000000, Actual_Vol3: -1355.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11519.110834, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.782909, Absolute Angle: 1.654130, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027950, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

663: 28184 [INFO] CHASSIS_PID_TURN, Time: 8388, Actual_Vol1: -1842.000000, Actual_Vol2: 1879.000000, Actual_Vol3: -1435.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11471.280196, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.783064, Absolute Angle: 1.654132, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027950, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000

664: 28184 [INFO] CHASSIS_PID_TURN, Time: 8398, Actual_Vol1: -1854.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1454.000000, Actual_Vol4: 1275.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11423.425110, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.785509, Absolute Angle: 1.654175, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027010

8, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
665: 28184 [INFO] CHASSIS_PID_TURN, Time: 8408, Actual_Vol1: -1860.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1460.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11375.607290, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.781782, Absolute Angle: 1.654110, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026898, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
666: 28186 [INFO] CHASSIS_PID_TURN, Time: 8418, Actual_Vol1: -1928.000000, Actual_Vol2: 1947.000000, Actual_Vol3: -1454.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11327.823362, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.778393, Absolute Angle: 1.654051, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030287, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
667: 28186 [INFO] CHASSIS_PID_TURN, Time: 8428, Actual_Vol1: -1965.000000, Actual_Vol2: 1977.000000, Actual_Vol3: -1577.000000, Actual_Vol4: 1324.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11280.088308, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.773505, Absolute Angle: 1.653965, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.033844, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
668: 28186 [INFO] CHASSIS_PID_TURN, Time: 8438, Actual_Vol1: -1977.000000, Actual_Vol2: 1953.000000, Actual_Vol3: -1589.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11232.375681, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.771263, Absolute Angle: 1.653926, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031460, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
669: 28188 [INFO] CHASSIS_PID_TURN, Time: 8448, Actual_Vol1: -2088.000000, Actual_Vol2: 1959.000000, Actual_Vol3: -1589.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11184.709671, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.766601, Absolute Angle: 1.653845, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030403, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
670: 28188 [INFO] CHASSIS_PID_TURN, Time: 8458, Actual_Vol1: -2218.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1737.000000, Actual_Vol4: 1423.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11137.064974, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.764470, Absolute Angle: 1.653808, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032534, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
671: 28188 [INFO] CHASSIS_PID_TURN, Time: 8468, Actual_Vol1: -2242.000000, Actual_Vol2: 2008.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11089.393564, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.767141, Absolute Angle: 1.653854, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032534, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
672: 28190 [INFO] CHASSIS_PID_TURN, Time: 8478, Actual_Vol1: -2248.000000, Actual_Vol2: 1996.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 1448.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11041.753320, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.764024, Absolute Angle: 1.653800, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031665, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
673: 28190 [INFO] CHASSIS_PID_TURN, Time: 8488, Actual_Vol1: -2261.000000, Actual_Vol2: 2107.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 1478.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10994.139135, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.761419, Absolute Angle: 1.653755, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032593, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
674: 28190 [INFO] CHASSIS_PID_TURN, Time: 8498, Actual_Vol1: -2248.000000, Actual_Vol2: 2236.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10946.616701, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.752243, Absolute Angle: 1.653594, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041768, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
675: 28192 [INFO] CHASSIS_PID_TURN, Time: 8508, Actual_Vol1: -2255.000000, Actual_Vol2: 2267.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10899.215497, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.740120, Absolute Angle: 1.653383, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.053891, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -121.400000, Actual_Vel4: 18.800000
676: 28192 [INFO] CHASSIS_PID_TURN, Time: 8518, Actual_Vol1: -2322.000000, Actual_Vol2: 2335.000000, Actual_Vol3: -1793.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10851.935681, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.727982, Absolute Angle: 1.653171, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.066030, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -121.400000, Actual_Vel4: 18.800000
677: 28192 [INFO] CHASSIS_PID_TURN, Time: 8528, Actual_Vol1: -2378.000000, Actual_Vol2: 2384.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10804.730562, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.720512, Absolute Angle: 1.653041, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.073500, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -121.400000, Actual_Vel4: 8.000000
678: 28194 [INFO] CHASSIS_PID_TURN, Time: 8538, Actual_Vol1: -2365.000000, Actual_Vol2: 2396.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 1571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10757.693002, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.703756, Absolute Angle: 1.652748, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.090256, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -53.800000, Actual_Vel4: 8.000000
679: 28194 [INFO] CHASSIS_PID_TURN, Time: 8548, Actual_Vol1: -2378.000000, Actual_Vol2: 2445.000000, Actual_Vol3: -1786.000000, Actual_Vol4: 1540.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10710.854184, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.683882, Absolute Angle: 1.652401, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.105038, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 74.400000, Actual_Vel3: -53.800000, Actual_Vel4: 9.400000
680: 28194 [INFO] CHASSIS_PID_TURN, Time: 8558, Actual_Vol1: -2384.000000, Actual_Vol2: 2445.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 1540.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10664.258594, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.659559, Absolute Angle: 1.651977, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.125950, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 74.400000, Actual_Vel3: -53.800000, Actual_Vel4: 9.400000
681: 28196 [INFO] CHASSIS_PID_TURN, Time: 8568, Actual_Vol1: -2384.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 1565.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10618.902689, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.535590, Absolute Angle: 1.649813, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.249918, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 74.400000, Actual_Vel3: -53.800000, Actual_Vel4: 9.400000
682: 28196 [INFO] CHASSIS_PID_TURN, Time: 8578, Actual_Vol1: -2384.000000, Actual_Vol2: 2409.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10573.802672, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.510002, Absolute Angle: 1.649366, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.275507, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 74.400000, Actual_Vel3: -53.800000, Actual_Vel4: 9.400000
683: 28196 [INFO] CHASSIS_PID_TURN, Time: 8588, Actual_Vol1: -2415.000000, Actual_Vol2: 2409.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10529.004126, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.479855, Absolute Angle: 1.648840, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.305654, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 44.600000, Actual_Vel3: -53.800000, Actual_Vel4: 14.000000
684: 28198 [INFO] CHASSIS_PID_TURN, Time: 8598, Actual_Vol1: -2427.000000, Actual_Vol2: 2409.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 1657.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10484.548980, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.445515, Absolute Angle: 1.648241, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.336267, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 44.600000, Actual_Vel3: -53.800000, Actual_Vel4: 14.000000
685: 28198 [INFO] CHASSIS_PID_TURN, Time: 8608, Actual_Vol1: -2384.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10440.434188, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.411479, Absolute Angle: 1.647647, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.366914, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 44.600000, Actual_Vel3: -32.600000, Actual_Vel4: 14.000000
686: 28198 [INFO] CHASSIS_PID_TURN, Time: 8618, Actual_Vol1: -2427.000000, Actual_Vol2: 2445.000000, Actual_Vol3: -1879.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10396.593890, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.384030, Absolute Angle: 1.647168, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.389476, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 44.600000, Actual_Vel3: -32.600000, Actual_Vel4: 14.000000
687: 28200 [INFO] CHASSIS_PID_TURN, Time: 8628, Actual_Vol1: -2445.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 1645.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10352.954411, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.363948, Absolute Angle: 1.646817, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.407315, over slew: 0, Actual_Vel1: -11.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 14.000000
688: 28200 [INFO] CHASSIS_PID_TURN, Time: 8638, Actual_Vol1: -2427.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1885.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10309.553997, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.340041, Absolute Angle: 1.646400, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.427100, over slew: 0, Actual_Vel1: -11.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 14.000000
689: 28200 [INFO] CHASSIS_PID_TURN, Time: 8648, Actual_Vol1: -2384.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 1682.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10266.362055, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.319194, Absolute Angle: 1.646036, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.447947, over slew: 0, Actual_Vel1: -11.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 15.200000
690: 28202 [INFO] CHASSIS_PID_TURN, Time: 8658, Actual_Vol1: -2390.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1589.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10223.301003, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.306105, Absolute Angle: 1.645808, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.461036, over slew: 0, Actual_Vel1: -11.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 15.200000
691: 28202 [INFO] CHASSIS_PID_TURN, Time: 8668, Actual_Vol1: -2421.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10180.404155, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.289685, Absolute Angle: 1.645521, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.474340, over slew: 0, Actual_Vel1: -16.000000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 15.200000
692: 28202 [INFO] CHASSIS_PID_TURN, Time: 8678, Actual_Vol1: -2464.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10137.679945, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.272421, Absolute Angle: 1.645220, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.488998, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 18.400000
693: 28204 [INFO] CHASSIS_PID_TURN, Time: 8688, Actual_Vol1: -2384.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 1749.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10095.174294, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.250565, Absolute Angle: 1.644838, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.501678, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 15.400000, Actual_Vel3: -11.200000, Actual_Vel4: 18.400000
694: 28204 [INFO] CHASSIS_PID_TURN, Time: 8698, Actual_Vol1: -2378.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1768.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10052.936666, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.223763, Absolute Angle: 1.644371, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.516358, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 15.400000, Actual_Vel3: -11.200000, Actual_Vel4: 15.000000
695: 28204 [INFO] CHASSIS_PID_TURN, Time: 8708, Actual_Vol1: -2445.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1688.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10010.960418, kD: 35.00000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.197625, Absolute Angle: 1.643914, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.533057, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 8.400000, Actual_Vel3: -11.200000, Actual_Vel4: 15.000000
696: 28206 [INFO] CHASSIS_PID_TURN, Time: 8718, Actual_Vol1: -2476.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1682.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9969.283510, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.167691, Absolute Angle: 1.643392, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.552821, over slew: 0, Actual_Vel1: -15.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 15.000000
697: 28206 [INFO] CHASSIS_PID_TURN, Time: 8728, Actual_Vol1: -2482.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9927.929555, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.135396, Absolute Angle: 1.642828, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.568360, over slew: 0, Actual_Vel1: -15.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 9.400000
698: 28206 [INFO] CHASSIS_PID_TURN, Time: 8738, Actual_Vol1: -2470.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9886.872932, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.105662, Absolute Angle: 1.642309, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.578220, over slew: 0, Actual_Vel1: -15.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 9.400000
699: 28208 [INFO] CHASSIS_PID_TURN, Time: 8748, Actual_Vol1: -2482.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1565.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9846.058266, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.081467, Absolute Angle: 1.641887, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.578092, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 9.400000
700: 28208 [INFO] CHASSIS_PID_TURN, Time: 8758, Actual_Vol1: -2476.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9805.406102, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 94.065216, Absolute Angle: 1.641603, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.470374, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 12.000000
701: 28208 [INFO] CHASSIS_PID_TURN, Time: 8768, Actual_Vol1: -2470.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9765.884228, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 93.952187, Absolute Angle: 1.639631, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.557814, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 12.000000
702: 28210 [INFO] CHASSIS_PID_TURN, Time: 8778, Actual_Vol1: -2470.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9726.507020, kD: 35.000000

, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.937721, Absolute Angle: 1.639378, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.542134 , over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 8.800000, Actual_Vel3: -30.800000, Actual_Vel4: 12.000000

703: 28210 [INFO] CHASSIS_PID_TURN, Time: 8788, Actual_Vol1: -2464.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9688.250017, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825700, Absolute Angle: 1.637423, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.619814 , over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 9.200000, Actual_Vel3: -30.800000, Actual_Vel4: 12.000000

704: 28210 [INFO] CHASSIS_PID_TURN, Time: 8798, Actual_Vol1: -1423.000000, Actual_Vol2: 2470.000000, Actual_Vol3: -2014.000000, Actual_Vol4: 1491.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9650.108517, kD: 35.000000 , kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.814150, Absolute Angle: 1.637222, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.597329 , over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 9.200000, Actual_Vel3: -30.800000, Actual_Vel4: 14.600000

705: 28212 [INFO] CHASSIS_PID_TURN, Time: 8808, Actual_Vol1: -308.000000, Actual_Vol2: 610.000000, Actual_Vol3: -505.000000, Actual_Vol4: 326.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9612.111502, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.799702, Absolute Angle: 1.636969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.584328, ov er slew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 9.200000, Actual_Vel3: -30.800000, Actual_Vel4: 14.600000

706: 28212 [INFO] CHASSIS_PID_TURN, Time: 8818, Actual_Vol1: -209.000000, Actual_Vol2: 142.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9574.284811, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.782669, Absolute Angle: 1.636672, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.581279, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: -0.000000, Actual_Vel4: 7.000000

707: 28212 [INFO] CHASSIS_PID_TURN, Time: 8828, Actual_Vol1: -376.000000, Actual_Vol2: 142.000000, Actual_Vol3: -18.000000, Actual_Vol4: 74.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9536.572311, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.771250, Absolute Angle: 1.636473, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.568791, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 7.000000

708: 28214 [INFO] CHASSIS_PID_TURN, Time: 8838, Actual_Vol1: -431.000000, Actual_Vol2: 290.000000, Actual_Vol3: -86.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9498.876339, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.769597, Absolute Angle: 1.636444, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.549597, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 7.000000

709: 28214 [INFO] CHASSIS_PID_TURN, Time: 8848, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -86.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9461.090957, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.778538, Absolute Angle: 1.636600, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.536508, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 7.000000

710: 28214 [INFO] CHASSIS_PID_TURN, Time: 8858, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -105.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9423.068570, kD: 35.000000, kI: 0. 000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.802239, Absolute Angle: 1.637014, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.520088, over slew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 6.200000

711: 28216 [INFO] CHASSIS_PID_TURN, Time: 8868, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9384.852787, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.821578, Absolute Angle: 1.637351, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.502824, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 6.200000

712: 28216 [INFO] CHASSIS_PID_TURN, Time: 8878, Actual_Vol1: -431.000000, Actual_Vol2: 376.000000, Actual_Vol3: -99.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9346.463682, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.838910, Absolute Angle: 1.637654, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.480968, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -0.000000, Actual_Vel3: 13.800000, Actual_Vel4: 6.200000

713: 28216 [INFO] CHASSIS_PID_TURN, Time: 8888, Actual_Vol1: -431.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9308.043660, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.842002, Absolute Angle: 1.637708, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.454165, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -0.000000, Actual_Vel3: 13.800000, Actual_Vel4: 0.000000

714: 28218 [INFO] CHASSIS_PID_TURN, Time: 8898, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9269.638012, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.840565, Absolute Angle: 1.637683, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.428028, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -9.800000, Actual_Vel3: 13.800000, Actual_Vel4: 0.000000

715: 28218 [INFO] CHASSIS_PID_TURN, Time: 8908, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9231.228000, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.841001, Absolute Angle: 1.637690, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.398094, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -9.800000, Actual_Vel3: 13.800000, Actual_Vel4: 0.000000

716: 28218 [INFO] CHASSIS_PID_TURN, Time: 8918, Actual_Vol1: -474.000000, Actual_Vol2: 407.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9192.822930, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.840507, Absolute Angle: 1.637682, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.365798, over s lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -9.800000, Actual_Vel3: 13.800000, Actual_Vel4: -10.000000

717: 28220 [INFO] CHASSIS_PID_TURN, Time: 8928, Actual_Vol1: -511.000000, Actual_Vol2: 444.000000, Actual_Vol3: -92.000000, Actual_Vol4: 99.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9154.528393, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829454, Absolute Angle: 1.637489, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.336065, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 10.400000, Actual_Vel4: -10.000000

718: 28220 [INFO] CHASSIS_PID_TURN, Time: 8938, Actual_Vol1: -517.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9116.278227, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825017, Absolute Angle: 1.637411, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.311869, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 10.400000, Actual_Vel4: -10.000000

719: 28220 [INFO] CHASSIS_PID_TURN, Time: 8948, Actual_Vol1: -517.000000, Actual_Vol2: 450.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9078.024683, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825354, Absolute Angle: 1.637417, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.295619, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 10.400000, Actual_Vel4: 0.000000

720: 28222 [INFO] CHASSIS_PID_TURN, Time: 8958, Actual_Vol1: -524.000000, Actual_Vol2: 493.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9039.695346, kD: 35.000000, kI: 0.0 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.832934, Absolute Angle: 1.637549, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.182590, over s lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

721: 28222 [INFO] CHASSIS_PID_TURN, Time: 8968, Actual_Vol1: -567.000000, Actual_Vol2: 530.000000, Actual_Vol3: -203.000000, Actual_Vol4: 197.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9001.363272, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833207, Absolute Angle: 1.637554, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.168124, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

722: 28222 [INFO] CHASSIS_PID_TURN, Time: 8978, Actual_Vol1: -598.000000, Actual_Vol2: 536.000000, Actual_Vol3: -228.000000, Actual_Vol4: 216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8963.066825, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829645, Absolute Angle: 1.637492, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

723: 28224 [INFO] CHASSIS_PID_TURN, Time: 8988, Actual_Vol1: -616.000000, Actual_Vol2: 542.000000, Actual_Vol3: -228.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8924.735179, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833165, Absolute Angle: 1.637553, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

724: 28224 [INFO] CHASSIS_PID_TURN, Time: 8998, Actual_Vol1: -616.000000, Actual_Vol2: 536.000000, Actual_Vol3: -228.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8886.404676, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833050, Absolute Angle: 1.637551, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

725: 28224 [INFO] CHASSIS_PID_TURN, Time: 9008, Actual_Vol1: -659.000000, Actual_Vol2: 579.000000, Actual_Vol3: -234.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8848.145580, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825910, Absolute Angle: 1.637427, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

726: 28226 [INFO] CHASSIS_PID_TURN, Time: 9018, Actual_Vol1: -690.000000, Actual_Vol2: 628.000000, Actual_Vol3: -290.000000, Actual_Vol4: 302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8809.848196, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829738, Absolute Angle: 1.637494, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

727: 28226 [INFO] CHASSIS_PID_TURN, Time: 9028, Actual_Vol1: -702.000000, Actual_Vol2: 634.000000, Actual_Vol3: -302.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8771.511791, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833640, Absolute Angle: 1.637562, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

728: 28226 [INFO] CHASSIS_PID_TURN, Time: 9038, Actual_Vol1: -702.000000, Actual_Vol2: 634.000000, Actual_Vol3: -314.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8733.135983, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.837581, Absolute Angle: 1.637631, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.063464, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

729: 28228 [INFO] CHASSIS_PID_TURN, Time: 9048, Actual_Vol1: -825.000000, Actual_Vol2: 671.000000, Actual_Vol3: -394.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8694.792686, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.834330, Absolute Angle: 1.637574, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.039764, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

730: 28228 [INFO] CHASSIS_PID_TURN, Time: 9058, Actual_Vol1: -899.000000, Actual_Vol2: 715.000000, Actual_Vol3: -425.000000, Actual_Vol4: 413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8656.407600, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.838509, Absolute Angle: 1.637647, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020424, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

731: 28228 [INFO] CHASSIS_PID_TURN, Time: 9068, Actual_Vol1: -918.000000, Actual_Vol2: 758.000000, Actual_Vol3: -431.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8618.022995, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.838460, Absolute Angle: 1.637646, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016986, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

732: 28230 [INFO] CHASSIS_PID_TURN, Time: 9078, Actual_Vol1: -924.000000, Actual_Vol2: 758.000000, Actual_Vol3: -431.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8579.668747, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.835425, Absolute Angle: 1.637593, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016986, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

733: 28230 [INFO] CHASSIS_PID_TURN, Time: 9088, Actual_Vol1: -973.000000, Actual_Vol2: 838.000000, Actual_Vol3: -474.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8541.350433, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.831831, Absolute Angle: 1.637530, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015985, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

734: 28230 [INFO] CHASSIS_PID_TURN, Time: 9098, Actual_Vol1: -1016.000000, Actual_Vol2: 918.000000, Actual_Vol3: -517.000000, Actual_Vol4: 499.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8503.000448, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.834999, Absolute Angle: 1.637585, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015985, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

735: 28232 [INFO] CHASSIS_PID_TURN, Time: 9108, Actual_Vol1: -1023.000000, Actual_Vol2: 942.000000, Actual_Vol3: -524.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8464.651458, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.834899, Absolute Angle: 1.637584, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015490, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

736: 28232 [INFO] CHASSIS_PID_TURN, Time: 9118, Actual_Vol1: -1023.000000, Actual_Vol2: 942.000000, Actual_Vol3: -517.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8426.368258, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.828320, Absolute Angle: 1.637469, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013492, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

737: 28232 [INFO] CHASSIS_PID_TURN, Time: 9128, Actual_Vol1: -1066.000000, Actual_Vol2: 992.000000, Actual_Vol3: -567.000000, Actual_Vol4: 524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8388.125158, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.824310, Absolute Angle: 1.637399, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, ov er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

738: 28234 [INFO] CHASSIS_PID_TURN, Time: 9138, Actual_Vol1: -1109.000000, Actual_Vol2: 1029.000000, Actual_Vol3: -610.000000, Actual_Vol4: 591.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8349.880613, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.824455, Absolute Angle: 1.637401, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

739: 28234 [INFO] CHASSIS_PID_TURN, Time: 9148, Actual_Vol1: -1146.000000, Actual_Vol2: 1035.000000, Actual_Vol3: -567.000000, Actual_Vol4: 616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8311.605127, kD: 35.000000, k I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.827549, Absolute Angle: 1.637455, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

740: 28234 [INFO] CHASSIS_PID_TURN, Time: 9158, Actual_Vol1: -1152.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -530.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8273.282705, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.832242, Absolute Angle: 1.637537, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

741: 28236 [INFO] CHASSIS_PID_TURN, Time: 9168, Actual_Vol1: -1195.000000, Actual_Vol2: 1084.000000, Actual_Vol3: -573.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8234.952886, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.832982, Absolute Angle: 1.637550, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

742: 28236 [INFO] CHASSIS_PID_TURN, Time: 9178, Actual_Vol1: -1238.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -610.000000, Actual_Vol4: 696.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8196.613676, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833921, Absolute Angle: 1.637567, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

743: 28236 [INFO] CHASSIS_PID_TURN, Time: 9188, Actual_Vol1: -1244.000000, Actual_Vol2: 1164.000000, Actual_Vol3: -616.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8158.281592, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833208, Absolute Angle: 1.637554, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

744: 28238 [INFO] CHASSIS_PID_TURN, Time: 9198, Actual_Vol1: -1244.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -622.000000, Actual_Vol4: 715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8119.942452, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833914, Absolute Angle: 1.637567, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

745: 28238 [INFO] CHASSIS_PID_TURN, Time: 9208, Actual_Vol1: -1294.000000, Actual_Vol2: 1207.000000, Actual_Vol3: -659.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8081.592165, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.835029, Absolute Angle: 1.637586, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

746: 28238 [INFO] CHASSIS_PID_TURN, Time: 9218, Actual_Vol1: -1343.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -696.000000, Actual_Vol4: 887.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8043.263630, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.832854, Absolute Angle: 1.637548, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

747: 28240 [INFO] CHASSIS_PID_TURN, Time: 9228, Actual_Vol1: -1337.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -708.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8004.978617, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.828501, Absolute Angle: 1.637472, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

748: 28240 [INFO] CHASSIS_PID_TURN, Time: 9238, Actual_Vol1: -1343.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -708.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7966.739377, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.823924, Absolute Angle: 1.637392, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014585, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

749: 28240 [INFO] CHASSIS_PID_TURN, Time: 9248, Actual_Vol1: -1380.000000, Actual_Vol2: 1300.000000, Actual_Vol3: -832.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7928.478380, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.826100, Absolute Angle: 1.637430, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014585, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

750: 28242 [INFO] CHASSIS_PID_TURN, Time: 9258, Actual_Vol1: -1429.000000, Actual_Vol2: 1349.000000, Actual_Vol3: -912.000000, Actual_Vol4: 1004.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7890.183104, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829528, Absolute Angle: 1.637490, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014536, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

751: 28242 [INFO] CHASSIS_PID_TURN, Time: 9268, Actual_Vol1: -1435.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -930.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7851.886783, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829632, Absolute Angle: 1.637492, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011501, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000

752: 28242 [INFO] CHASSIS_PID_TURN, Time: 9278, Actual_Vol1: -1435.000000, Actual_Vol2: 1361.000000, Actual_Vol3: -930.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7813.620635, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.826615, Absolute Angle: 1.637439, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011105, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

753: 28244 [INFO] CHASSIS_PID_TURN, Time: 9288, Actual_Vol1: -1478.000000, Actual_Vol2: 1398.000000, Actual_Vol3: -973.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7775.369359, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825128, Absolute Angle: 1.637413, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011105, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

754: 28244 [INFO] CHASSIS_PID_TURN, Time: 9298, Actual_Vol1: -1571.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1016.000000, Actual_Vol4: 1096.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7737.115916, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825344, Absolute Angle: 1.637417, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011105, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

755: 28244 [INFO] CHASSIS_PID_TURN, Time: 9308, Actual_Vol1: -1577.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 1146.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7698.915362, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.820055, Absolute Angle: 1.637325, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014973, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

756: 28246 [INFO] CHASSIS_PID_TURN, Time: 9318, Actual_Vol1: -1577.000000, Actual_Vol2: 1460.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 1047.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7660.746233, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.816913, Absolute Angle: 1.637270, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018116, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

757: 28246 [INFO] CHASSIS_PID_TURN, Time: 9328, Actual_Vol1: -1620.000000, Actual_Vol2: 1540.000000, Actual_Vol3: -1072.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7622.538661, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.820757, Absolute Angle: 1.637337, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018116, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

758: 28246 [INFO] CHASSIS_PID_TURN, Time: 9338, Actual_Vol1: -1657.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -1146.000000, Actual_Vol4: 1103.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7584.335960, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.820270, Absolute Angle: 1.637328, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018116, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

759: 28248 [INFO] CHASSIS_PID_TURN, Time: 9348, Actual_Vol1: -1669.000000, Actual_Vol2: 1589.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1146.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7546.201214, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.813475, Absolute Angle: 1.637210, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021554, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

760: 28248 [INFO] CHASSIS_PID_TURN, Time: 9358, Actual_Vol1: -1663.000000, Actual_Vol2: 1632.000000, Actual_Vol3: -1201.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7508.152231, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.804898, Absolute Angle: 1.637060, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030130, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

761: 28248 [INFO] CHASSIS_PID_TURN, Time: 9368, Actual_Vol1: -1712.000000, Actual_Vol2: 1669.000000, Actual_Vol3: -1244.000000, Actual_Vol4: 1226.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7470.125382, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.802685, Absolute Angle: 1.637021, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032344, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

762: 28250 [INFO] CHASSIS_PID_TURN, Time: 9378, Actual_Vol1: -1743.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 1244.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7432.099997, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.802539, Absolute Angle: 1.637019, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032490, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

763: 28250 [INFO] CHASSIS_PID_TURN, Time: 9388, Actual_Vol1: -1756.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7394.159516, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.794048, Absolute Angle: 1.636871, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040981, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

764: 28250 [INFO] CHASSIS_PID_TURN, Time: 9398, Actual_Vol1: -1793.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1300.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7356.253926, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.790559, Absolute Angle: 1.636810, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.044470, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

765: 28252 [INFO] CHASSIS_PID_TURN, Time: 9408, Actual_Vol1: -1842.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1337.000000, Actual_Vol4: 1324.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7318.380692, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.787323, Absolute Angle: 1.636753, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.045530, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

766: 28252 [INFO] CHASSIS_PID_TURN, Time: 9418, Actual_Vol1: -1854.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1343.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7280.530860, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.784983, Absolute Angle: 1.636712, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.044649, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000

767: 28252 [INFO] CHASSIS_PID_TURN, Time: 9428, Actual_Vol1: -1860.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1349.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7242.752476, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.777838, Absolute Angle: 1.636588, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051794, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

768: 28254 [INFO] CHASSIS_PID_TURN, Time: 9438, Actual_Vol1: -1922.000000, Actual_Vol2: 1731.000000, Actual_Vol3: -1386.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7204.964227, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.778825, Absolute Angle: 1.636605, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051794, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

769: 28254 [INFO] CHASSIS_PID_TURN, Time: 9448, Actual_Vol1: -1965.000000, Actual_Vol2: 1774.000000, Actual_Vol3: -1441.000000, Actual_Vol4: 1423.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7167.167834, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.779639, Absolute Angle: 1.636619, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051794, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

770: 28254 [INFO] CHASSIS_PID_TURN, Time: 9458, Actual_Vol1: -1971.000000, Actual_Vol2: 1774.000000, Actual_Vol3: -1448.000000, Actual_Vol4: 1441.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7129.412958, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.775488, Absolute Angle: 1.636547, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.054145, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

771: 28256 [INFO] CHASSIS_PID_TURN, Time: 9468, Actual_Vol1: -1965.000000, Actual_Vol2: 1817.000000, Actual_Vol3: -1540.000000, Actual_Vol4: 1441.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7091.709969, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.770299, Absolute Angle: 1.636456, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.056316, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

772: 28256 [INFO] CHASSIS_PID_TURN, Time: 9478, Actual_Vol1: -2082.000000, Actual_Vol2: 1866.000000, Actual_Vol3: -1583.000000, Actual_Vol4: 1558.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7054.005175, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.770479, Absolute Angle: 1.636459, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.055045, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

773: 28256 [INFO] CHASSIS_PID_TURN, Time: 9488, Actual_Vol1: -2144.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1583.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7016.319773, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.768540, Absolute Angle: 1.636426, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.056804, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

774: 28258 [INFO] CHASSIS_PID_TURN, Time: 9498, Actual_Vol1: -2051.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1595.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6978.689419, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.763035, Absolute Angle: 1.636329, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057722, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

775: 28258 [INFO] CHASSIS_PID_TURN, Time: 9508, Actual_Vol1: -2119.000000, Actual_Vol2: 1947.000000, Actual_Vol3: -1675.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6941.090534, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.759889, Absolute Angle: 1.636275, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.060869, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

776: 28258 [INFO] CHASSIS_PID_TURN, Time: 9548, Actual_Vol1: -2372.000000, Actual_Vol2: 2242.000000, Actual_Vol3: -1749.000000, Actual_Vol4: 1854.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6791.222618, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.740561, Absolute Angle: 1.635937, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.064337, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 0.000000

777: 28262 [INFO] CHASSIS_PID_TURN, Time: 9558, Actual_Vol1: -2384.000000, Actual_Vol2: 2267.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 1873.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6753.883455, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.733916, Absolute Angle: 1.635821, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.068769

, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 0.000000
778: 28262 [INFO] CHASSIS_PID_TURN, Time: 9568, Actual_Vol1: -2384.000000, Actual_Vol2: 2341.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 1866.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6716.564862, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.731859, Absolute Angle: 1.635785, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.070679, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 0.000000
779: 28262 [INFO] CHASSIS_PID_TURN, Time: 9578, Actual_Vol1: -2445.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1811.000000, Actual_Vol4: 2039.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6679.228762, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.733610, Absolute Angle: 1.635816, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.062189, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 7.200000
780: 28264 [INFO] CHASSIS_PID_TURN, Time: 9588, Actual_Vol1: -2464.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 1934.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6641.997507, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.723126, Absolute Angle: 1.635633, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.067433, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 7.200000
781: 28264 [INFO] CHASSIS_PID_TURN, Time: 9598, Actual_Vol1: -2470.000000, Actual_Vol2: 2409.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 1879.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6604.854088, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.714342, Absolute Angle: 1.635480, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072982, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
782: 28264 [INFO] CHASSIS_PID_TURN, Time: 9608, Actual_Vol1: -2464.000000, Actual_Vol2: 2452.000000, Actual_Vol3: -1934.000000, Actual_Vol4: 1780.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6567.803698, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.705039, Absolute Angle: 1.635317, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.079944, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
783: 28266 [INFO] CHASSIS_PID_TURN, Time: 9618, Actual_Vol1: -2458.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 1836.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6530.813847, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.698985, Absolute Angle: 1.635212, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.080654, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
784: 28266 [INFO] CHASSIS_PID_TURN, Time: 9628, Actual_Vol1: -2458.000000, Actual_Vol2: 2464.000000, Actual_Vol3: -2033.000000, Actual_Vol4: 1866.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6494.001859, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.681199, Absolute Angle: 1.634901, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098441, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 40.800000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
785: 28266 [INFO] CHASSIS_PID_TURN, Time: 9638, Actual_Vol1: -2464.000000, Actual_Vol2: 2452.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1959.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6457.294575, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.670728, Absolute Angle: 1.634718, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.108911, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 40.800000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
786: 28268 [INFO] CHASSIS_PID_TURN, Time: 9648, Actual_Vol1: -2464.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2088.000000, Actual_Vol4: 2039.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6420.712424, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 93.658215, Absolute Angle: 1.634500, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.117273, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 40.800000, Actual_Vel3: -14.400000, Actual_Vel4: 8.800000
787: 28268 [INFO] CHASSIS_PID_TURN, Time: 9658, Actual_Vol1: -2464.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2094.000000, Actual_Vol4: 2057.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6385.265999, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 93.544642, Absolute Angle: 1.632518, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.225837, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 40.800000, Actual_Vel3: -14.400000, Actual_Vel4: 10.000000
788: 28268 [INFO] CHASSIS_PID_TURN, Time: 9668, Actual_Vol1: -2464.000000, Actual_Vol2: 2464.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1996.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6350.005535, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 93.526046, Absolute Angle: 1.632193, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.244433, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 24.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.000000
789: 28270 [INFO] CHASSIS_PID_TURN, Time: 9678, Actual_Vol1: -2599.000000, Actual_Vol2: 2452.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1990.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6314.960514, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 93.504502, Absolute Angle: 1.631817, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.264038, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 24.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.000000
790: 28270 [INFO] CHASSIS_PID_TURN, Time: 9688, Actual_Vol1: -2636.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 2002.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6280.120484, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.484003, Absolute Angle: 1.631459, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.279032, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
791: 28270 [INFO] CHASSIS_PID_TURN, Time: 9698, Actual_Vol1: -2649.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -2088.000000, Actual_Vol4: 1811.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6246.528634, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.359185, Absolute Angle: 1.629281, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.400704, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
792: 28272 [INFO] CHASSIS_PID_TURN, Time: 9708, Actual_Vol1: -2753.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -2051.000000, Actual_Vol4: 1774.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6213.235072, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.329356, Absolute Angle: 1.628760, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.425598, over slew: 0, Actual_Vel1: -97.600000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
793: 28272 [INFO] CHASSIS_PID_TURN, Time: 9718, Actual_Vol1: -2698.000000, Actual_Vol2: 2507.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1848.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6180.200307, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.303476, Absolute Angle: 1.628309, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.443752, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
794: 28272 [INFO] CHASSIS_PID_TURN, Time: 9728, Actual_Vol1: -2513.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2045.000000, Actual_Vol4: 1959.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6147.399769, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.280054, Absolute Angle: 1.627900, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.463993, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
795: 28274 [INFO] CHASSIS_PID_TURN, Time: 9738, Actual_Vol1: -2482.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1983.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6114.828538, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.257123, Absolute Angle: 1.627500, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.483438, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 24.800000, Actual_Vel3: -21.800000, Actual_Vel4: 16.800000
796: 28274 [INFO] CHASSIS_PID_TURN, Time: 9748, Actual_Vol1: -2612.000000, Actual_Vol2: 2569.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1805.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6082.525094, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.230344, Absolute Angle: 1.627032, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.503572, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 24.800000, Actual_Vel3: -21.800000, Actual_Vel4: 16.800000
797: 28274 [INFO] CHASSIS_PID_TURN, Time: 9758, Actual_Vol1: -2643.000000, Actual_Vol2: 2643.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1947.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6050.522356, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.200274, Absolute Angle: 1.626507, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.533336, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 12.800000, Actual_Vel3: -21.800000, Actual_Vel4: 16.800000
798: 28276 [INFO] CHASSIS_PID_TURN, Time: 9768, Actual_Vol1: -2636.000000, Actual_Vol2: 2593.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1983.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6018.795426, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.172693, Absolute Angle: 1.626026, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.560917, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 12.800000, Actual_Vel3: -37.000000, Actual_Vel4: 20.800000
799: 28276 [INFO] CHASSIS_PID_TURN, Time: 9778, Actual_Vol1: -2649.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1879.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5987.352611, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.144281, Absolute Angle: 1.625530, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.578844, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 12.800000, Actual_Vel3: -37.000000, Actual_Vel4: 20.800000
800: 28276 [INFO] CHASSIS_PID_TURN, Time: 9788, Actual_Vol1: -2649.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2088.000000, Actual_Vol4: 1793.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5956.168541, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.118407, Absolute Angle: 1.625079, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.595935, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 12.800000, Actual_Vel3: -37.000000, Actual_Vel4: 20.800000
801: 28278 [INFO] CHASSIS_PID_TURN, Time: 9798, Actual_Vol1: -2655.000000, Actual_Vol2: 2636.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1762.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5925.250205, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.091834, Absolute Angle: 1.624615, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.613205, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.000000, Actual_Vel3: -37.000000, Actual_Vel4: 20.800000
802: 28278 [INFO] CHASSIS_PID_TURN, Time: 9808, Actual_Vol1: -2643.000000, Actual_Vol2: 2661.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1940.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5895.534956, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.971525, Absolute Angle: 1.622515, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.727460, over slew: 0, Actual_Vel1: -13.200000, Actual_Vel2: 10.000000, Actual_Vel3: -38.200000, Actual_Vel4: 20.800000
803: 28278 [INFO] CHASSIS_PID_TURN, Time: 9818, Actual_Vol1: -2519.000000, Actual_Vol2: 1608.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1977.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5866.000235, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.953472, Absolute Angle: 1.622200, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.727727, over slew: 0, Actual_Vel1: -13.200000, Actual_Vel2: 10.000000, Actual_Vel3: -38.200000, Actual_Vel4: 42.400000
804: 28280 [INFO] CHASSIS_PID_TURN, Time: 9828, Actual_Vol1: -690.000000, Actual_Vol2: 388.000000, Actual_Vol3: -524.000000, Actual_Vol4: 468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5836.624652, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.937558, Absolute Angle: 1.621922, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.733170, over slew: 0, Actual_Vel1: -13.200000, Actual_Vel2: -0.000000, Actual_Vel3: -38.200000, Actual_Vel4: 42.400000
805: 28280 [INFO] CHASSIS_PID_TURN, Time: 9838, Actual_Vol1: -154.000000, Actual_Vol2: 179.000000, Actual_Vol3: -105.000000, Actual_Vol4: 191.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5807.398728, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.922592, Absolute Angle: 1.621661, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.735623, over slew: 0, Actual_Vel1: -5.000000, Actual_Vel2: -0.000000, Actual_Vel3: -8.800000, Actual_Vel4: 9.600000
806: 28280 [INFO] CHASSIS_PID_TURN, Time: 9848, Actual_Vol1: -376.000000, Actual_Vol2: 234.000000, Actual_Vol3: -25.000000, Actual_Vol4: 25.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5778.172805, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.922592, Absolute Angle: 1.621661, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.622050, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 9.600000
807: 28282 [INFO] CHASSIS_PID_TURN, Time: 9858, Actual_Vol1: -480.000000, Actual_Vol2: 320.000000, Actual_Vol3: -80.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5748.850784, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.932202, Absolute Angle: 1.621829, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.603454, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 9.600000
808: 28282 [INFO] CHASSIS_PID_TURN, Time: 9868, Actual_Vol1: -511.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5719.286856, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.956393, Absolute Angle: 1.622251, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.581910, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 9.600000
809: 28282 [INFO] CHASSIS_PID_TURN, Time: 9878, Actual_Vol1: -517.000000, Actual_Vol2: 370.000000, Actual_Vol3: -92.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5689.438042, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984881, Absolute Angle: 1.622748, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.561411, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 4.000000, Actual_Vel4: 0.000000
810: 28284 [INFO] CHASSIS_PID_TURN, Time: 9888, Actual_Vol1: -524.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5659.361393, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.007665, Absolute Angle: 1.623146, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.436593, over slew: 0, Actual_Vel1: 10.200000, Actual_Vel2: -179.800000, Actual_Vel3: 4.000000, Actual_Vel4: 0.000000
811: 28284 [INFO] CHASSIS_PID_TURN, Time: 9898, Actual_Vol1: -517.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5629.179734, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.018166, Absolute Angle: 1.623329, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.406764, over slew: 0, Actual_Vel1: 10.200000, Actual_Vel2: -179.800000, Actual_Vel3: 4.000000, Actual_Vel4: 0.000000
812: 28284 [INFO] CHASSIS_PID_TURN, Time: 9908, Actual_Vol1: -517.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5598.977349, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.020239, Absolute Angle: 1.623365, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.380884, over slew: 0, Actual_Vel1: 10.200000, Actual_Vel2: -179.800000, Actual_Vel3: 9.400000, Actual_Vel4: -17.800000
813: 28286 [INFO] CHASSIS_PID_TURN, Time: 9918, Actual_Vol1: -524.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5568.800803, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.017655, Absolute Angle: 1.623320, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.357461, over slew: 0, Actual_Vel1: 45.000000, Actual_Vel2: -179.800000, Actual_Vel3: 9.400000, Actual_Vel4: -17.800000
814: 28286 [INFO] CHASSIS_PID_TURN, Time: 9928, Actual_Vol1: -524.000000, Actual_Vol2: 370.000000, Actual_Vol3: -92.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5538.670154, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.013065, Absolute Angle: 1.623240, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.334531, over slew: 0, Actual_Vel1: 45.000000, Actual_Vel2: -179.800000, Actual_Vel3: 9.400000, Actual_Vel4: -17.800000
815: 28286 [INFO] CHASSIS_PID_TURN, Time: 9938, Actual_Vol1: -517.000000, Actual_Vol2: 400.000000, Actual_Vol3: -92.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5508.659824, kD: 35.000000, kI: 0.00

0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.001033, Absolute Angle: 1.623030, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.307752, over slew: 0, Actual_Vel1: 45.000000, Actual_Vel2: -179.800000, Actual_Vel3: 15.800000, Actual_Vel4: -17.800000

816: 28288 [INFO] CHASSIS_PID_TURN, Time: 9948, Actual_Vol1: -591.000000, Actual_Vol2: 444.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5478.744807, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.991502, Absolute Angle: 1.622864, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.277681, over slew: 0, Actual_Vel1: 24.200000, Actual_Vel2: -179.800000, Actual_Vel3: 15.800000, Actual_Vel4: -17.800000

817: 28288 [INFO] CHASSIS_PID_TURN, Time: 9958, Actual_Vol1: -610.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5448.801152, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.994365, Absolute Angle: 1.622914, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.250101, over slew: 0, Actual_Vel1: 24.200000, Actual_Vel2: -179.800000, Actual_Vel3: 15.800000, Actual_Vel4: -17.800000

818: 28288 [INFO] CHASSIS_PID_TURN, Time: 9968, Actual_Vol1: -604.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5418.822919, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.997823, Absolute Angle: 1.622974, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.221689, over slew: 0, Actual_Vel1: 24.200000, Actual_Vel2: -179.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

819: 28290 [INFO] CHASSIS_PID_TURN, Time: 9978, Actual_Vol1: -616.000000, Actual_Vol2: 493.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5388.816034, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.000689, Absolute Angle: 1.623024, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.195815, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

820: 28290 [INFO] CHASSIS_PID_TURN, Time: 9988, Actual_Vol1: -678.000000, Actual_Vol2: 536.000000, Actual_Vol3: -203.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5358.829322, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.998671, Absolute Angle: 1.622989, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.169241, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

821: 28290 [INFO] CHASSIS_PID_TURN, Time: 9998, Actual_Vol1: -702.000000, Actual_Vol2: 536.000000, Actual_Vol3: -216.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5328.843241, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.998608, Absolute Angle: 1.622988, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

822: 28292 [INFO] CHASSIS_PID_TURN, Time: 10008, Actual_Vol1: -702.000000, Actual_Vol2: 542.000000, Actual_Vol3: -222.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5298.832414, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.001083, Absolute Angle: 1.623031, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

823: 28292 [INFO] CHASSIS_PID_TURN, Time: 10018, Actual_Vol1: -696.000000, Actual_Vol2: 579.000000, Actual_Vol3: -222.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5268.782984, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004943, Absolute Angle: 1.623098, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

824: 28292 [INFO] CHASSIS_PID_TURN, Time: 10028, Actual_Vol1: -875.000000, Actual_Vol2: 628.000000, Actual_Vol3: -290.000000, Actual_Vol4: 197.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5238.708014, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.007497, Absolute Angle: 1.623143, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

825: 28294 [INFO] CHASSIS_PID_TURN, Time: 10038, Actual_Vol1: -912.000000, Actual_Vol2: 634.000000, Actual_Vol3: -302.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5208.611632, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.009638, Absolute Angle: 1.623180, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

826: 28294 [INFO] CHASSIS_PID_TURN, Time: 10048, Actual_Vol1: -918.000000, Actual_Vol2: 641.000000, Actual_Vol3: -308.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5178.544147, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.006749, Absolute Angle: 1.623130, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.088036, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

827: 28294 [INFO] CHASSIS_PID_TURN, Time: 10058, Actual_Vol1: -918.000000, Actual_Vol2: 678.000000, Actual_Vol3: -308.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5148.438763, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.010538, Absolute Angle: 1.623196, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.063846, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

828: 28296 [INFO] CHASSIS_PID_TURN, Time: 10068, Actual_Vol1: -992.000000, Actual_Vol2: 715.000000, Actual_Vol3: -413.000000, Actual_Vol4: 296.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5118.346091, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.009267, Absolute Angle: 1.623174, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035357, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

829: 28296 [INFO] CHASSIS_PID_TURN, Time: 10078, Actual_Vol1: -1016.000000, Actual_Vol2: 764.000000, Actual_Vol3: -431.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5088.216881, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.012921, Absolute Angle: 1.623237, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028737, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

830: 28296 [INFO] CHASSIS_PID_TURN, Time: 10088, Actual_Vol1: -1023.000000, Actual_Vol2: 758.000000, Actual_Vol3: -444.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5058.053537, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.016334, Absolute Angle: 1.623297, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028737, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

831: 28298 [INFO] CHASSIS_PID_TURN, Time: 10098, Actual_Vol1: -1023.000000, Actual_Vol2: 844.000000, Actual_Vol3: -431.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5027.903553, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.014998, Absolute Angle: 1.623274, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028737, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

832: 28298 [INFO] CHASSIS_PID_TURN, Time: 10108, Actual_Vol1: -1096.000000, Actual_Vol2: 930.000000, Actual_Vol3: -431.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4997.752478, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.015107, Absolute Angle: 1.623276, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026153, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

833: 28298 [INFO] CHASSIS_PID_TURN, Time: 10118, Actual_Vol1: -1140.000000, Actual_Vol2: 942.000000, Actual_Vol3: -511.000000, Actual_Vol4: 394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4967.607023, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.014545, Absolute Angle: 1.623266, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024833, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

834: 28300 [INFO] CHASSIS_PID_TURN, Time: 10128, Actual_Vol1: -1152.000000, Actual_Vol2: 949.000000, Actual_Vol3: -517.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4937.507901, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.009912, Absolute Angle: 1.623185, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024833, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

835: 28300 [INFO] CHASSIS_PID_TURN, Time: 10138, Actual_Vol1: -1146.000000, Actual_Vol2: 992.000000, Actual_Vol3: -524.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4907.455496, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.005241, Absolute Angle: 1.623103, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024833, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

836: 28300 [INFO] CHASSIS_PID_TURN, Time: 10148, Actual_Vol1: -1226.000000, Actual_Vol2: 1035.000000, Actual_Vol3: -524.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4877.416506, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.003899, Absolute Angle: 1.623080, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021969, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

837: 28302 [INFO] CHASSIS_PID_TURN, Time: 10158, Actual_Vol1: -1244.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -598.000000, Actual_Vol4: 480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4847.394118, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.002239, Absolute Angle: 1.623051, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018511, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

838: 28302 [INFO] CHASSIS_PID_TURN, Time: 10168, Actual_Vol1: -1238.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -610.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4817.399702, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.999442, Absolute Angle: 1.623002, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017726, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

839: 28302 [INFO] CHASSIS_PID_TURN, Time: 10178, Actual_Vol1: -1250.000000, Actual_Vol2: 1084.000000, Actual_Vol3: -616.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4787.380155, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.001955, Absolute Angle: 1.623046, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017726, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

840: 28304 [INFO] CHASSIS_PID_TURN, Time: 10188, Actual_Vol1: -1312.000000, Actual_Vol2: 1164.000000, Actual_Vol3: -616.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4757.346402, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.003375, Absolute Angle: 1.623071, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017726, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

841: 28304 [INFO] CHASSIS_PID_TURN, Time: 10198, Actual_Vol1: -1337.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -684.000000, Actual_Vol4: 573.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4727.299867, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004654, Absolute Angle: 1.623093, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

842: 28304 [INFO] CHASSIS_PID_TURN, Time: 10208, Actual_Vol1: -1324.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -708.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4697.212153, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.008771, Absolute Angle: 1.623165, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

843: 28306 [INFO] CHASSIS_PID_TURN, Time: 10218, Actual_Vol1: -1337.000000, Actual_Vol2: 1214.000000, Actual_Vol3: -702.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4667.167192, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004496, Absolute Angle: 1.623090, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

844: 28306 [INFO] CHASSIS_PID_TURN, Time: 10228, Actual_Vol1: -1411.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -708.000000, Actual_Vol4: 628.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4637.126608, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004058, Absolute Angle: 1.623083, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

845: 28306 [INFO] CHASSIS_PID_TURN, Time: 10238, Actual_Vol1: -1423.000000, Actual_Vol2: 1269.000000, Actual_Vol3: -887.000000, Actual_Vol4: 665.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4607.050075, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.007653, Absolute Angle: 1.623146, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

846: 28308 [INFO] CHASSIS_PID_TURN, Time: 10248, Actual_Vol1: -1435.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -924.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4577.001531, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004854, Absolute Angle: 1.623097, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

847: 28308 [INFO] CHASSIS_PID_TURN, Time: 10258, Actual_Vol1: -1435.000000, Actual_Vol2: 1306.000000, Actual_Vol3: -924.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4546.978236, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.002330, Absolute Angle: 1.623053, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

848: 28308 [INFO] CHASSIS_PID_TURN, Time: 10268, Actual_Vol1: -1546.000000, Actual_Vol2: 1343.000000, Actual_Vol3: -930.000000, Actual_Vol4: 795.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4516.981087, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.999715, Absolute Angle: 1.623007, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

849: 28310 [INFO] CHASSIS_PID_TURN, Time: 10278, Actual_Vol1: -1565.000000, Actual_Vol2: 1349.000000, Actual_Vol3: -1004.000000, Actual_Vol4: 887.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4487.001906, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.997918, Absolute Angle: 1.622976, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018416, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

850: 28310 [INFO] CHASSIS_PID_TURN, Time: 10288, Actual_Vol1: -1577.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -1023.000000, Actual_Vol4: 924.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4457.083575, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.991833, Absolute Angle: 1.622869, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023274, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

851: 28310 [INFO] CHASSIS_PID_TURN, Time: 10298, Actual_Vol1: -1577.000000, Actual_Vol2: 1398.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4427.166534, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.991704, Absolute Angle: 1.622867, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023403, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

852: 28312 [INFO] CHASSIS_PID_TURN, Time: 10308, Actual_Vol1: -1651.000000, Actual_Vol2: 1441.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4397.216054, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.995048, Absolute Angle: 1.622926, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022841, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

853: 28312 [INFO] CHASSIS_PID_TURN, Time: 10318, Actual_Vol1: -1669.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1096.000000, Actual_Vol4: 992.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4367.297690, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.991836, Absolute Angle: 1.622869, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018208, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

854: 28312 [INFO] CHASSIS_PID_TURN, Time: 10328, Actual_Vol1: -1675.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4337.347626, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.995006, Absolute Angle: 1.622925, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

855: 28314 [INFO] CHASSIS_PID_TURN, Time: 10338, Actual_Vol1: -1675.000000, Actual_Vol2: 1534.000000, Actual_Vol3: -1047.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4307.376159, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.997147, Absolute Angle: 1.622962, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

856: 28314 [INFO] CHASSIS_PID_TURN, Time: 10348, Actual_Vol1: -1743.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -1103.000000, Actual_Vol4: 1072.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4277.369182, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.000698, Absolute Angle: 1.623024, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

857: 28314 [INFO] CHASSIS_PID_TURN, Time: 10358, Actual_Vol1: -1756.000000, Actual_Vol2: 1595.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1146.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4247.381476, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.998771, Absolute Angle: 1.622991, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

858: 28316 [INFO] CHASSIS_PID_TURN, Time: 10368, Actual_Vol1: -1756.000000, Actual_Vol2: 1589.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1158.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4217.443767, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.993771, Absolute Angle: 1.622903, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

859: 28316 [INFO] CHASSIS_PID_TURN, Time: 10378, Actual_Vol1: -1799.000000, Actual_Vol2: 1632.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4187.541115, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.990265, Absolute Angle: 1.622842, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018506, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

860: 28316 [INFO] CHASSIS_PID_TURN, Time: 10388, Actual_Vol1: -1848.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1232.000000, Actual_Vol4: 1195.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4157.609994, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.993112, Absolute Angle: 1.622892, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018506, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

861: 28318 [INFO] CHASSIS_PID_TURN, Time: 10398, Actual_Vol1: -1854.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1244.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4127.704872, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.990512, Absolute Angle: 1.622846, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018506, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

862: 28318 [INFO] CHASSIS_PID_TURN, Time: 10408, Actual_Vol1: -1854.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4097.847226, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.985765, Absolute Angle: 1.622764, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021889, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

863: 28318 [INFO] CHASSIS_PID_TURN, Time: 10418, Actual_Vol1: -1928.000000, Actual_Vol2: 1719.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4068.000169, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984706, Absolute Angle: 1.622745, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022948, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

864: 28320 [INFO] CHASSIS_PID_TURN, Time: 10428, Actual_Vol1: -1965.000000, Actual_Vol2: 1768.000000, Actual_Vol3: -1331.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4038.215873, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.978430, Absolute Angle: 1.622635, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.029224, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

865: 28320 [INFO] CHASSIS_PID_TURN, Time: 10438, Actual_Vol1: -1940.000000, Actual_Vol2: 1768.000000, Actual_Vol3: -1343.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4008.369396, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984648, Absolute Angle: 1.622744, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026425, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000

866: 28320 [INFO] CHASSIS_PID_TURN, Time: 10448, Actual_Vol1: -1866.000000, Actual_Vol2: 1780.000000, Actual_Vol3: -1355.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3978.523195, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984620, Absolute Angle: 1.622744, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023900, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

867: 28322 [INFO] CHASSIS_PID_TURN, Time: 10458, Actual_Vol1: -1934.000000, Actual_Vol2: 1817.000000, Actual_Vol3: -1355.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3948.677935, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984526, Absolute Angle: 1.622742, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022268, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

868: 28322 [INFO] CHASSIS_PID_TURN, Time: 10468, Actual_Vol1: -1965.000000, Actual_Vol2: 1866.000000, Actual_Vol3: -1429.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3918.827226, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.985071, Absolute Angle: 1.622751, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022268, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

869: 28322 [INFO] CHASSIS_PID_TURN, Time: 10478, Actual_Vol1: -1977.000000, Actual_Vol2: 1879.000000, Actual_Vol3: -1448.000000, Actual_Vol4: 1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3889.000529, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.982670, Absolute Angle: 1.622709, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022268, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

870: 28324 [INFO] CHASSIS_PID_TURN, Time: 10488, Actual_Vol1: -2101.000000, Actual_Vol2: 1953.000000, Actual_Vol3: -1441.000000, Actual_Vol4: 1441.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3859.224650, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.977588, Absolute Angle: 1.622621, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023110, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

871: 28324 [INFO] CHASSIS_PID_TURN, Time: 10498, Actual_Vol1: -2224.000000, Actual_Vol2: 1983.000000, Actual_Vol3: -1558.000000, Actual_Vol4: 1485.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3829.483416, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.974123, Absolute Angle: 1.622560, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026574, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

872: 28324 [INFO] CHASSIS_PID_TURN, Time: 10508, Actual_Vol1: -2261.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1577.000000, Actual_Vol4: 1571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3799.753510, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.972991, Absolute Angle: 1.622541, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027707, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

873: 28326 [INFO] CHASSIS_PID_TURN, Time: 10518, Actual_Vol1: -2267.000000, Actual_Vol2: 1959.000000, Actual_Vol3: -1577.000000, Actual_Vol4: 1540.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3770.060069, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.969344, Absolute Angle: 1.622477, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031354, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: 98.600000, Actual_Vel3: -0.000000, Actual_Vel4: 218.200000

874: 28326 [INFO] CHASSIS_PID_TURN, Time: 10528, Actual_Vol1: -2335.000000, Actual_Vol2: 1953.000000, Actual_Vol3: -1577.000000, Actual_Vol4: 1454.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3740.411832, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.964824, Absolute Angle: 1.622398, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035874, over slew: 0, Actual_Vel1: -11.800000, Actual_Vel2: 98.600000, Actual_Vel3: -0.000000, Actual_Vel4: 218.200000

875: 28326 [INFO] CHASSIS_PID_TURN, Time: 10538, Actual_Vol1: -2341.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1725.000000, Actual_Vol4: 1485.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3710.841625, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.957021, Absolute Angle: 1.622262, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.043677, over slew: 0, Actual_Vel1: -11.800000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

876: 28328 [INFO] CHASSIS_PID_TURN, Time: 10548, Actual_Vol1: -2267.000000, Actual_Vol2: 1996.000000, Actual_Vol3: -1626.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3681.358118, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.948351, Absolute Angle: 1.622111, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.050420, over slew: 0, Actual_Vel1: -11.800000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

877: 28328 [INFO] CHASSIS_PID_TURN, Time: 10558, Actual_Vol1: -2335.000000, Actual_Vol2: 2107.000000, Actual_Vol3: -1595.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3651.954690, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.940343, Absolute Angle: 1.621971, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.053428, over slew: 0, Actual_Vel1: -11.800000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

878: 28328 [INFO] CHASSIS_PID_TURN, Time: 10568, Actual_Vol1: -2372.000000, Actual_Vol2: 2230.000000, Actual_Vol3: -1725.000000, Actual_Vol4: 1657.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3622.606995, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.934770, Absolute Angle: 1.621873, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.058343, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

879: 28330 [INFO] CHASSIS_PID_TURN, Time: 10578, Actual_Vol1: -2372.000000, Actual_Vol2: 2261.000000, Actual_Vol3: -1756.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3593.359998, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.924700, Absolute Angle: 1.621698, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.068412, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

880: 28330 [INFO] CHASSIS_PID_TURN, Time: 10588, Actual_Vol1: -2378.000000, Actual_Vol2: 2267.000000, Actual_Vol3: -1756.000000, Actual_Vol4: 1756.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3564.150596, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.920940, Absolute Angle: 1.621632, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.069572, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

881: 28330 [INFO] CHASSIS_PID_TURN, Time: 10598, Actual_Vol1: -2409.000000, Actual_Vol2: 2341.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 1768.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3534.992946, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.915765, Absolute Angle: 1.621542, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.070000, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

882: 28332 [INFO] CHASSIS_PID_TURN, Time: 10608, Actual_Vol1: -2458.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 1805.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3505.924751, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.906819, Absolute Angle: 1.621386, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.078251, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

883: 28332 [INFO] CHASSIS_PID_TURN, Time: 10618, Actual_Vol1: -2470.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1885.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3476.934234, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.899052, Absolute Angle: 1.621250, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.086019, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

884: 28332 [INFO] CHASSIS_PID_TURN, Time: 10628, Actual_Vol1: -2470.000000, Actual_Vol2: 2433.000000, Actual_Vol3: -1934.000000, Actual_Vol4: 1928.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3447.939593, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.899464, Absolute Angle: 1.621257, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.086019, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

885: 28334 [INFO] CHASSIS_PID_TURN, Time: 10638, Actual_Vol1: -2470.000000, Actual_Vol2: 2470.000000, Actual_Vol3: -2020.000000, Actual_Vol4: 1971.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3418.920448, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.901914, Absolute Angle: 1.621300, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.086019, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000

886: 28334 [INFO] CHASSIS_PID_TURN, Time: 10648, Actual_Vol1: -2458.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 2051.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3389.921917, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.899853, Absolute Angle: 1.621264, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.086019, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -14.200000, Actual_Vel4: 218.200000

887: 28334 [INFO] CHASSIS_PID_TURN, Time: 10658, Actual_Vol1: -2470.000000, Actual_Vol2: 2482.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 2064.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3360.950346, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.897157, Absolute Angle: 1.621217, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.087914, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.200000, Actual_Vel4: 8.000000

888: 28336 [INFO] CHASSIS_PID_TURN, Time: 10668, Actual_Vol1: -2544.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 2027.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3332.051526, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.889882, Absolute Angle: 1.621090, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.092788, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.200000, Actual_Vel4: 8.000000

889: 28336 [INFO] CHASSIS_PID_TURN, Time: 10678, Actual_Vol1: -2624.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2057.000000, Actual_Vol4: 2020.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3303.204743, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.884678, Absolute Angle: 1.620999, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.092910, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 8.000000

890: 28336 [INFO] CHASSIS_PID_TURN, Time: 10688, Actual_Vol1: -2636.000000, Actual_Vol2: 2482.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 2064.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3274.465916, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.873883, Absolute Angle: 1.620811, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.100240

1, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 8.000000
891: 28338 [INFO] CHASSIS_PID_TURN, Time: 10698, Actual_Vol1: -2717.000000, Actual_Vol2: 2612.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 2039.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3245.901738, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.856418, Absolute Angle: 1.620506, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.11657
3, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 14.200000
892: 28338 [INFO] CHASSIS_PID_TURN, Time: 10708, Actual_Vol1: -2723.000000, Actual_Vol2: 2661.000000, Actual_Vol3: -2070.000000, Actual_Vol4: 2027.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3217.480992, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.842075, Absolute Angle: 1.620256, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.12726
9, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
893: 28338 [INFO] CHASSIS_PID_TURN, Time: 10718, Actual_Vol1: -2649.000000, Actual_Vol2: 2704.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 2020.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3189.241403, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.823959, Absolute Angle: 1.619939, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.14086
5, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
894: 28340 [INFO] CHASSIS_PID_TURN, Time: 10728, Actual_Vol1: -2643.000000, Actual_Vol2: 2544.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1983.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3161.121995, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.811941, Absolute Angle: 1.619730, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.14508
0, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
895: 28340 [INFO] CHASSIS_PID_TURN, Time: 10738, Actual_Vol1: -2704.000000, Actual_Vol2: 2643.000000, Actual_Vol3: -2070.000000, Actual_Vol4: 1977.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3133.175294, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 92.794670, Absolute Angle: 1.619428, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.15368
1, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
896: 28340 [INFO] CHASSIS_PID_TURN, Time: 10748, Actual_Vol1: -2797.000000, Actual_Vol2: 2667.000000, Actual_Vol3: -2070.000000, Actual_Vol4: 2008.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3106.339340, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 92.683595, Absolute Angle: 1.617490, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.25674
7, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
897: 28342 [INFO] CHASSIS_PID_TURN, Time: 10758, Actual_Vol1: -2846.000000, Actual_Vol2: 2704.000000, Actual_Vol3: -2070.000000, Actual_Vol4: 2064.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3079.633009, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 92.670633, Absolute Angle: 1.617263, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.26413
6, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
898: 28342 [INFO] CHASSIS_PID_TURN, Time: 10768, Actual_Vol1: -2846.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -2150.000000, Actual_Vol4: 2039.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3053.101534, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 92.653148, Absolute Angle: 1.616958, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.27155
2, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.200000, Actual_Vel3: -14.400000, Actual_Vel4: 21.000000
899: 28342 [INFO] CHASSIS_PID_TURN, Time: 10778, Actual_Vol1: -2858.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -2230.000000, Actual_Vol4: 1990.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3026.763809, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.633772, Absolute Angle: 1.616620, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.28716
8, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.200000, Actual_Vel3: -76.000000, Actual_Vel4: 17.000000
900: 28344 [INFO] CHASSIS_PID_TURN, Time: 10788, Actual_Vol1: -2864.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -2255.000000, Actual_Vol4: 1977.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3001.576437, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.518737, Absolute Angle: 1.614612, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.39702
8, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 9.000000, Actual_Vel3: -76.000000, Actual_Vel4: 17.000000
901: 28344 [INFO] CHASSIS_PID_TURN, Time: 10798, Actual_Vol1: -2852.000000, Actual_Vol2: 2544.000000, Actual_Vol3: -2255.000000, Actual_Vol4: 1971.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2976.553486, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.502295, Absolute Angle: 1.614325, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.40452
4, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 9.000000, Actual_Vel3: -76.000000, Actual_Vel4: 17.000000
902: 28344 [INFO] CHASSIS_PID_TURN, Time: 10808, Actual_Vol1: -2895.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2248.000000, Actual_Vol4: 1928.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2951.734750, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.481874, Absolute Angle: 1.613969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.42004
1, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 12.400000
903: 28346 [INFO] CHASSIS_PID_TURN, Time: 10818, Actual_Vol1: -2926.000000, Actual_Vol2: 2636.000000, Actual_Vol3: -2322.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2927.123046, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.461170, Absolute Angle: 1.613608, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.44074
4, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 13.200000
904: 28346 [INFO] CHASSIS_PID_TURN, Time: 10828, Actual_Vol1: -2938.000000, Actual_Vol2: 2661.000000, Actual_Vol3: -2378.000000, Actual_Vol4: 1731.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2902.697655, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.442539, Absolute Angle: 1.613282, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.45937
5, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 13.200000
905: 28346 [INFO] CHASSIS_PID_TURN, Time: 10838, Actual_Vol1: -2920.000000, Actual_Vol2: 2704.000000, Actual_Vol3: -2384.000000, Actual_Vol4: 1836.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2878.388887, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.430877, Absolute Angle: 1.613079, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.46897
6, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 9.800000, Actual_Vel3: -0.000000, Actual_Vel4: 13.200000
906: 28348 [INFO] CHASSIS_PID_TURN, Time: 10848, Actual_Vol1: -2852.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -2347.000000, Actual_Vol4: 1860.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2854.201928, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.418696, Absolute Angle: 1.612866, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.47846
1, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 9.800000, Actual_Vel3: -16.600000, Actual_Vel4: 13.200000
907: 28348 [INFO] CHASSIS_PID_TURN, Time: 10858, Actual_Vol1: -2889.000000, Actual_Vol2: 2778.000000, Actual_Vol3: -2279.000000, Actual_Vol4: 1725.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2830.123628, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.407830, Absolute Angle: 1.612677, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.48205
2, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 9.800000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
908: 28348 [INFO] CHASSIS_PID_TURN, Time: 10868, Actual_Vol1: -2932.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1595.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2806.213571, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.391006, Absolute Angle: 1.612383, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.49367
3, over slew: 0, Actual_Vel1: -20.000000, Actual_Vel2: 9.800000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
909: 28350 [INFO] CHASSIS_PID_TURN, Time: 10878, Actual_Vol1: -2895.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1700.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2782.513562, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.370001, Absolute Angle: 1.612016, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.50388
2, over slew: 0, Actual_Vel1: -20.000000, Actual_Vel2: 9.800000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
910: 28350 [INFO] CHASSIS_PID_TURN, Time: 10888, Actual_Vol1: -2858.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1823.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2759.986142, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.252742, Absolute Angle: 1.609970, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.60367
6, over slew: 0, Actual_Vel1: -20.000000, Actual_Vel2: 9.800000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
911: 28350 [INFO] CHASSIS_PID_TURN, Time: 10898, Actual_Vol1: -2852.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2267.000000, Actual_Vol4: 1842.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2737.676877, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.230927, Absolute Angle: 1.609589, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.61114
8, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 7.400000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
912: 28352 [INFO] CHASSIS_PID_TURN, Time: 10908, Actual_Vol1: -696.000000, Actual_Vol2: 702.000000, Actual_Vol3: -505.000000, Actual_Vol4: 382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2715.568874, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.210800, Absolute Angle: 1.609238, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.613159, over slew: 0, Actual_Vel1: -5.000000, Actual_Vel2: 7.400000, Actual_Vel3: -15.400000, Actual_Vel4: 7.200000
913: 28352 [INFO] CHASSIS_PID_TURN, Time: 10918, Actual_Vol1: -166.000000, Actual_Vol2: 209.000000, Actual_Vol3: -99.000000, Actual_Vol4: 74.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2693.628761, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.194011, Absolute Angle: 1.608945, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.617929, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.400000, Actual_Vel3: -15.400000, Actual_Vel4: 7.200000
914: 28352 [INFO] CHASSIS_PID_TURN, Time: 10928, Actual_Vol1: -277.000000, Actual_Vol2: 277.000000, Actual_Vol3: -62.000000, Actual_Vol4: 18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2671.717675, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.191109, Absolute Angle: 1.608894, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.603561, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: 7.200000
915: 28354 [INFO] CHASSIS_PID_TURN, Time: 10938, Actual_Vol1: -413.000000, Actual_Vol2: 425.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2649.722701, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.199497, Absolute Angle: 1.609041, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.492487, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: -5.200000
916: 28354 [INFO] CHASSIS_PID_TURN, Time: 10948, Actual_Vol1: -444.000000, Actual_Vol2: 450.000000, Actual_Vol3: -105.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2627.550259, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.217244, Absolute Angle: 1.609350, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.479524, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: -5.200000
917: 28354 [INFO] CHASSIS_PID_TURN, Time: 10958, Actual_Vol1: -437.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2605.145496, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.240476, Absolute Angle: 1.609756, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.462039, over slew: 0, Actual_Vel1: 4.400000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: -5.200000
918: 28356 [INFO] CHASSIS_PID_TURN, Time: 10968, Actual_Vol1: -437.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2582.489090, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.265641, Absolute Angle: 1.610195, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.442664, over slew: 0, Actual_Vel1: 4.400000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: -14.800000
919: 28356 [INFO] CHASSIS_PID_TURN, Time: 10978, Actual_Vol1: -444.000000, Actual_Vol2: 456.000000, Actual_Vol3: -86.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2559.674045, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.281505, Absolute Angle: 1.610472, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.327629, over slew: 0, Actual_Vel1: 4.400000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
920: 28356 [INFO] CHASSIS_PID_TURN, Time: 10988, Actual_Vol1: -437.000000, Actual_Vol2: 450.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2536.808132, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.286591, Absolute Angle: 1.610561, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.311187, over slew: 0, Actual_Vel1: 25.000000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
921: 28358 [INFO] CHASSIS_PID_TURN, Time: 10998, Actual_Vol1: -437.000000, Actual_Vol2: 444.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2513.908787, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.289934, Absolute Angle: 1.610619, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.290765, over slew: 0, Actual_Vel1: 25.000000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
922: 28358 [INFO] CHASSIS_PID_TURN, Time: 11008, Actual_Vol1: -437.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2491.052805, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.285598, Absolute Angle: 1.610543, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.270062, over slew: 0, Actual_Vel1: 25.000000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
923: 28358 [INFO] CHASSIS_PID_TURN, Time: 11018, Actual_Vol1: -431.000000, Actual_Vol2: 487.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2468.269011, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.278379, Absolute Angle: 1.610417, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.251430, over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
924: 28360 [INFO] CHASSIS_PID_TURN, Time: 11028, Actual_Vol1: -499.000000, Actual_Vol2: 536.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2445.586886, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268213, Absolute Angle: 1.610240, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.239768, over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -8.800000
925: 28360 [INFO] CHASSIS_PID_TURN, Time: 11038, Actual_Vol1: -530.000000, Actual_Vol2: 536.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2422.990649, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.259624, Absolute Angle: 1.610090, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.227587, over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -8.800000
926: 28360 [INFO] CHASSIS_PID_TURN, Time: 11048, Actual_Vol1: -524.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2400.375022, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.261563, Absolute Angle: 1.610124, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.216721, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 168.400000, Actual_Vel4: -8.800000
927: 28362 [INFO] CHASSIS_PID_TURN, Time: 11058, Actual_Vol1: -524.000000, Actual_Vol2: 579.000000, Actual_Vol3: -154.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2377.732496, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.264253, Absolute Angle: 1.610171, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.199897, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 168.400000, Actual_Vel4: 0.000000
928: 28362 [INFO] CHASSIS_PID_TURN, Time: 11068, Actual_Vol1: -598.000000, Actual_Vol2: 622.000000, Actual_Vol3: -222.000000, Actual_Vol4: 197.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2355.044444, kD: 35.000000, kI:

0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268805, Absolute Angle: 1.610250, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.178892, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

929: 28362 [INFO] CHASSIS_PID_TURN, Time: 11078, Actual_Vol1: -610.000000, Actual_Vol2: 628.000000, Actual_Vol3: -228.000000, Actual_Vol4: 216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2332.375137, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.266931, Absolute Angle: 1.610217, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

930: 28364 [INFO] CHASSIS_PID_TURN, Time: 11088, Actual_Vol1: -616.000000, Actual_Vol2: 634.000000, Actual_Vol3: -228.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2309.657884, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.271725, Absolute Angle: 1.610301, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

931: 28364 [INFO] CHASSIS_PID_TURN, Time: 11098, Actual_Vol1: -616.000000, Actual_Vol2: 671.000000, Actual_Vol3: -234.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2286.921791, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.273609, Absolute Angle: 1.610334, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

932: 28364 [INFO] CHASSIS_PID_TURN, Time: 11108, Actual_Vol1: -622.000000, Actual_Vol2: 721.000000, Actual_Vol3: -265.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2264.216998, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.270479, Absolute Angle: 1.610279, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

933: 28366 [INFO] CHASSIS_PID_TURN, Time: 11118, Actual_Vol1: -684.000000, Actual_Vol2: 752.000000, Actual_Vol3: -302.000000, Actual_Vol4: 296.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2241.562151, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.265485, Absolute Angle: 1.610192, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

934: 28366 [INFO] CHASSIS_PID_TURN, Time: 11128, Actual_Vol1: -702.000000, Actual_Vol2: 752.000000, Actual_Vol3: -308.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2218.878395, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268376, Absolute Angle: 1.610243, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.090437, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

935: 28366 [INFO] CHASSIS_PID_TURN, Time: 11138, Actual_Vol1: -702.000000, Actual_Vol2: 752.000000, Actual_Vol3: -308.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2196.144145, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.273425, Absolute Angle: 1.610331, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072690, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

936: 28368 [INFO] CHASSIS_PID_TURN, Time: 11148, Actual_Vol1: -696.000000, Actual_Vol2: 832.000000, Actual_Vol3: -382.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2173.422322, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.272182, Absolute Angle: 1.610309, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.049458, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

937: 28368 [INFO] CHASSIS_PID_TURN, Time: 11158, Actual_Vol1: -887.000000, Actual_Vol2: 918.000000, Actual_Vol3: -419.000000, Actual_Vol4: 419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2150.709711, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.271261, Absolute Angle: 1.610293, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030311, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

938: 28368 [INFO] CHASSIS_PID_TURN, Time: 11168, Actual_Vol1: -918.000000, Actual_Vol2: 942.000000, Actual_Vol3: -431.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2127.971100, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.273861, Absolute Angle: 1.610338, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030311, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

939: 28370 [INFO] CHASSIS_PID_TURN, Time: 11178, Actual_Vol1: -924.000000, Actual_Vol2: 949.000000, Actual_Vol3: -431.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2105.259838, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.271126, Absolute Angle: 1.610291, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030311, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

940: 28370 [INFO] CHASSIS_PID_TURN, Time: 11188, Actual_Vol1: -924.000000, Actual_Vol2: 986.000000, Actual_Vol3: -480.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2082.598550, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.266129, Absolute Angle: 1.610203, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030311, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

941: 28370 [INFO] CHASSIS_PID_TURN, Time: 11198, Actual_Vol1: -998.000000, Actual_Vol2: 1029.000000, Actual_Vol3: -511.000000, Actual_Vol4: 505.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2059.902346, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.269620, Absolute Angle: 1.610264, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025975, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

942: 28372 [INFO] CHASSIS_PID_TURN, Time: 11208, Actual_Vol1: -1016.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -517.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2037.188879, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.271347, Absolute Angle: 1.610295, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018756, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

943: 28372 [INFO] CHASSIS_PID_TURN, Time: 11218, Actual_Vol1: -1023.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -517.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2014.503224, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268565, Absolute Angle: 1.610246, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014237, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

944: 28372 [INFO] CHASSIS_PID_TURN, Time: 11228, Actual_Vol1: -1029.000000, Actual_Vol2: 1078.000000, Actual_Vol3: -561.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1991.829857, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.267337, Absolute Angle: 1.610225, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014237, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

945: 28374 [INFO] CHASSIS_PID_TURN, Time: 11238, Actual_Vol1: -1096.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -604.000000, Actual_Vol4: 598.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1969.141539, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268832, Absolute Angle: 1.610251, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012298, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

946: 28374 [INFO] CHASSIS_PID_TURN, Time: 11248, Actual_Vol1: -1146.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -604.000000, Actual_Vol4: 604.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1946.408679, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.273286, Absolute Angle: 1.610328, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.009608, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

947: 28374 [INFO] CHASSIS_PID_TURN, Time: 11258, Actual_Vol1: -1146.000000, Actual_Vol2: 1164.000000, Actual_Vol3: -610.000000, Actual_Vol4: 610.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1923.709406, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.269927, Absolute Angle: 1.610270, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008376, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

948: 28376 [INFO] CHASSIS_PID_TURN, Time: 11268, Actual_Vol1: -1152.000000, Actual_Vol2: 1201.000000, Actual_Vol3: -659.000000, Actual_Vol4: 616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1901.031231, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.267818, Absolute Angle: 1.610233, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008376, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

949: 28376 [INFO] CHASSIS_PID_TURN, Time: 11278, Actual_Vol1: -1220.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -696.000000, Actual_Vol4: 684.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1878.344409, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268682, Absolute Angle: 1.610248, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008376, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

950: 28376 [INFO] CHASSIS_PID_TURN, Time: 11288, Actual_Vol1: -1238.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -702.000000, Actual_Vol4: 819.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1855.659189, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268522, Absolute Angle: 1.610245, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008376, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

951: 28378 [INFO] CHASSIS_PID_TURN, Time: 11298, Actual_Vol1: -1250.000000, Actual_Vol2: 1269.000000, Actual_Vol3: -696.000000, Actual_Vol4: 653.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1833.012143, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.264705, Absolute Angle: 1.610179, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.009157, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

952: 28378 [INFO] CHASSIS_PID_TURN, Time: 11308, Actual_Vol1: -1244.000000, Actual_Vol2: 1300.000000, Actual_Vol3: -819.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1810.374272, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.263787, Absolute Angle: 1.610163, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.010074, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

953: 28378 [INFO] CHASSIS_PID_TURN, Time: 11318, Actual_Vol1: -1318.000000, Actual_Vol2: 1343.000000, Actual_Vol3: -912.000000, Actual_Vol4: 801.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1787.691954, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268232, Absolute Angle: 1.610240, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.010074, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

954: 28380 [INFO] CHASSIS_PID_TURN, Time: 11328, Actual_Vol1: -1343.000000, Actual_Vol2: 1361.000000, Actual_Vol3: -936.000000, Actual_Vol4: 838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1765.031208, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.266075, Absolute Angle: 1.610203, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.010074, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

955: 28380 [INFO] CHASSIS_PID_TURN, Time: 11338, Actual_Vol1: -1337.000000, Actual_Vol2: 1361.000000, Actual_Vol3: -924.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1742.410826, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.262038, Absolute Angle: 1.610132, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011823, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

956: 28380 [INFO] CHASSIS_PID_TURN, Time: 11348, Actual_Vol1: -1386.000000, Actual_Vol2: 1404.000000, Actual_Vol3: -967.000000, Actual_Vol4: 832.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1719.776579, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.263425, Absolute Angle: 1.610156, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011823, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

957: 28382 [INFO] CHASSIS_PID_TURN, Time: 11358, Actual_Vol1: -1429.000000, Actual_Vol2: 1441.000000, Actual_Vol3: -1016.000000, Actual_Vol4: 912.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1697.137121, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.263946, Absolute Angle: 1.610165, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011823, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

958: 28382 [INFO] CHASSIS_PID_TURN, Time: 11368, Actual_Vol1: -1435.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 924.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1674.502773, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.263435, Absolute Angle: 1.610156, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011248, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

959: 28382 [INFO] CHASSIS_PID_TURN, Time: 11378, Actual_Vol1: -1435.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1651.858273, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.264450, Absolute Angle: 1.610174, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011248, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

960: 28384 [INFO] CHASSIS_PID_TURN, Time: 11388, Actual_Vol1: -1472.000000, Actual_Vol2: 1546.000000, Actual_Vol3: -1066.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1629.192063, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.266621, Absolute Angle: 1.610212, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011248, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

961: 28384 [INFO] CHASSIS_PID_TURN, Time: 11398, Actual_Vol1: -1546.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -1096.000000, Actual_Vol4: 973.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1606.534072, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.265799, Absolute Angle: 1.610198, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011248, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

962: 28384 [INFO] CHASSIS_PID_TURN, Time: 11408, Actual_Vol1: -1571.000000, Actual_Vol2: 1552.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1583.920735, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.261334, Absolute Angle: 1.610120, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000

963: 28386 [INFO] CHASSIS_PID_TURN, Time: 11418, Actual_Vol1: -1571.000000, Actual_Vol2: 1472.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1561.298266, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.262247, Absolute Angle: 1.610136, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

964: 28386 [INFO] CHASSIS_PID_TURN, Time: 11428, Actual_Vol1: -1614.000000, Actual_Vol2: 1546.000000, Actual_Vol3: -1201.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1538.708417, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.258985, Absolute Angle: 1.610079, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014301, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

965: 28386 [INFO] CHASSIS_PID_TURN, Time: 11438, Actual_Vol1: -1657.000000, Actual_Vol2: 1583.000000, Actual_Vol3: -1244.000000, Actual_Vol4: 1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1516.136456, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.257196, Absolute Angle: 1.610048, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016090, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

966: 28388 [INFO] CHASSIS_PID_TURN, Time: 11448, Actual_Vol1: -1663.000000, Actual_Vol2: 1583.000000, Actual_Vol3: -1244.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1493.587058, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.254940, Absolute Angle: 1.610008, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014987, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

967: 28388 [INFO] CHASSIS_PID_TURN, Time: 11458, Actual_Vol1: -1669.000000, Actual_Vol2: 1583.000000, Actual_Vol3: -1214.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1471.005361, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.258170, Absolute Angle: 1.610065, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013742, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

968: 28388 [INFO] CHASSIS_PID_TURN, Time: 11468, Actual_Vol1: -1712.000000, Actual_Vol2: 1626.000000, Actual_Vol3: -1207.000000, Actual_Vol4: 1146.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1448.421418, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.258394, Absolute Angle: 1.610068, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013742, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

969: 28390 [INFO] CHASSIS_PID_TURN, Time: 11478, Actual_Vol1: -1756.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1238.000000, Actual_Vol4: 1195.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1425.873969, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.254745, Absolute Angle: 1.610005, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013777, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

970: 28390 [INFO] CHASSIS_PID_TURN, Time: 11488, Actual_Vol1: -1762.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1250.000000, Actual_Vol4: 1238.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1403.360266, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.251370, Absolute Angle: 1.609946, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016862, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

971: 28390 [INFO] CHASSIS_PID_TURN, Time: 11498, Actual_Vol1: -1768.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1250.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1380.809447, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.255082, Absolute Angle: 1.610011, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016862, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

972: 28392 [INFO] CHASSIS_PID_TURN, Time: 11508, Actual_Vol1: -1805.000000, Actual_Vol2: 1731.000000, Actual_Vol3: -1300.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1358.268386, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.254106, Absolute Angle: 1.609994, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016862, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

973: 28392 [INFO] CHASSIS_PID_TURN, Time: 11518, Actual_Vol1: -1848.000000, Actual_Vol2: 1768.000000, Actual_Vol3: -1331.000000, Actual_Vol4: 1294.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1335.707196, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.256119, Absolute Angle: 1.610029, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015251, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

974: 28392 [INFO] CHASSIS_PID_TURN, Time: 11528, Actual_Vol1: -1854.000000, Actual_Vol2: 1780.000000, Actual_Vol3: -1343.000000, Actual_Vol4: 1331.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1313.180007, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.252719, Absolute Angle: 1.609969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015251, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

975: 28394 [INFO] CHASSIS_PID_TURN, Time: 11538, Actual_Vol1: -1860.000000, Actual_Vol2: 1780.000000, Actual_Vol3: -1349.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1290.687918, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.249209, Absolute Angle: 1.609908, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017412, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

976: 28394 [INFO] CHASSIS_PID_TURN, Time: 11548, Actual_Vol1: -1922.000000, Actual_Vol2: 1817.000000, Actual_Vol3: -1386.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1268.195764, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.249215, Absolute Angle: 1.609908, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017412, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

977: 28394 [INFO] CHASSIS_PID_TURN, Time: 11558, Actual_Vol1: -1965.000000, Actual_Vol2: 1866.000000, Actual_Vol3: -1435.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1245.734440, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.246132, Absolute Angle: 1.609854, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020489, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

978: 28396 [INFO] CHASSIS_PID_TURN, Time: 11568, Actual_Vol1: -1971.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1448.000000, Actual_Vol4: 1429.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1223.333931, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.240051, Absolute Angle: 1.609748, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026570, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000

979: 28396 [INFO] CHASSIS_PID_TURN, Time: 11578, Actual_Vol1: -1965.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1441.000000, Actual_Vol4: 1429.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1200.952173, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.238176, Absolute Angle: 1.609716, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028445, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

980: 28396 [INFO] CHASSIS_PID_TURN, Time: 11588, Actual_Vol1: -2088.000000, Actual_Vol2: 1940.000000, Actual_Vol3: -1478.000000, Actual_Vol4: 1429.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1178.568052, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.238412, Absolute Angle: 1.609720, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027623, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

981: 28398 [INFO] CHASSIS_PID_TURN, Time: 11598, Actual_Vol1: -2218.000000, Actual_Vol2: 1983.000000, Actual_Vol3: -1571.000000, Actual_Vol4: 1478.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1156.181603, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.238645, Absolute Angle: 1.609724, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024071, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

982: 28398 [INFO] CHASSIS_PID_TURN, Time: 11608, Actual_Vol1: -2168.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1571.000000, Actual_Vol4: 1565.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1133.829209, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.235239, Absolute Angle: 1.609664, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027008, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

983: 28398 [INFO] CHASSIS_PID_TURN, Time: 11618, Actual_Vol1: -2039.000000, Actual_Vol2: 2002.000000, Actual_Vol3: -1583.000000, Actual_Vol4: 1571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1111.496136, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.233307, Absolute Angle: 1.609631, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025678, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

984: 28400 [INFO] CHASSIS_PID_TURN, Time: 11628, Actual_Vol1: -2101.000000, Actual_Vol2: 2119.000000, Actual_Vol3: -1663.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1089.213579, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.228256, Absolute Angle: 1.609542, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030139, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

985: 28400 [INFO] CHASSIS_PID_TURN, Time: 11638, Actual_Vol1: -2230.000000, Actual_Vol2: 2248.000000, Actual_Vol3: -1749.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1066.968559, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.224502, Absolute Angle: 1.609477, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.033892, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

986: 28400 [INFO] CHASSIS_PID_TURN, Time: 11648, Actual_Vol1: -2242.000000, Actual_Vol2: 2267.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1044.768404, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.220015, Absolute Angle: 1.609399, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.038379, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

987: 28402 [INFO] CHASSIS_PID_TURN, Time: 11658, Actual_Vol1: -2248.000000, Actual_Vol2: 2273.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1022.596289, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.217212, Absolute Angle: 1.609350, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041183, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

988: 28402 [INFO] CHASSIS_PID_TURN, Time: 11668, Actual_Vol1: -2316.000000, Actual_Vol2: 2341.000000, Actual_Vol3: -1817.000000, Actual_Vol4: 1589.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1000.427246, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.216904, Absolute Angle: 1.609344, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.039215, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

989: 28402 [INFO] CHASSIS_PID_TURN, Time: 11678, Actual_Vol1: -2353.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 978.316339, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.211091, Absolute Angle: 1.609243, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.045028, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

990: 28404 [INFO] CHASSIS_PID_TURN, Time: 11688, Actual_Vol1: -2359.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 956.231250, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.208509, Absolute Angle: 1.609198, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.047610, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

991: 28404 [INFO] CHASSIS_PID_TURN, Time: 11698, Actual_Vol1: -2359.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1879.000000, Actual_Vol4: 1762.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 934.180666, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.205058, Absolute Angle: 1.609138, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051061, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

992: 28404 [INFO] CHASSIS_PID_TURN, Time: 11708, Actual_Vol1: -2402.000000, Actual_Vol2: 2470.000000, Actual_Vol3: -1940.000000, Actual_Vol4: 1762.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 912.161550, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.201912, Absolute Angle: 1.609083, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.054207, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

993: 28406 [INFO] CHASSIS_PID_TURN, Time: 11718, Actual_Vol1: -2445.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1805.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 890.112381, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.204917, Absolute Angle: 1.609135, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.050807, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

994: 28406 [INFO] CHASSIS_PID_TURN, Time: 11728, Actual_Vol1: -2452.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2033.000000, Actual_Vol4: 1885.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 868.094528, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.201785, Absolute Angle: 1.609080, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.047430, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

995: 28406 [INFO] CHASSIS_PID_TURN, Time: 11738, Actual_Vol1: -2452.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1965.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 846.084590, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.200994, Absolute Angle: 1.609067, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.048222, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

996: 28408 [INFO] CHASSIS_PID_TURN, Time: 11748, Actual_Vol1: -2452.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2094.000000, Actual_Vol4: 1971.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 824.063069, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.202152, Absolute Angle: 1.609087, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.045139, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

997: 28408 [INFO] CHASSIS_PID_TURN, Time: 11758, Actual_Vol1: -2464.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2088.000000, Actual_Vol4: 2020.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 802.022994, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.204007, Absolute Angle: 1.609119, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.039057, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

998: 28408 [INFO] CHASSIS_PID_TURN, Time: 11768, Actual_Vol1: -2464.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2094.000000, Actual_Vol4: 2070.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 780.021673, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.200132, Absolute Angle: 1.609052, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.038513, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000

999: 28410 [INFO] CHASSIS_PID_TURN, Time: 11778, Actual_Vol1: -2458.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2094.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 758.067472, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.195420, Absolute Angle: 1.608969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.043225, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1000: 28410 [INFO] CHASSIS_PID_TURN, Time: 11788, Actual_Vol1: -2532.000000, Actual_Vol2: 2636.000000, Actual_Vol3: -2174.000000, Actual_Vol4: 2082.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 736.118943, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.194853, Absolute Angle: 1.608959, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.043792, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1001: 28410 [INFO] CHASSIS_PID_TURN, Time: 11798, Actual_Vol1: -2618.000000, Actual_Vol2: 2661.000000, Actual_Vol3: -2255.000000, Actual_Vol4: 2082.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 714.179503, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.193944, Absolute Angle: 1.608944, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041295, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1002: 28412 [INFO] CHASSIS_PID_TURN, Time: 11808, Actual_Vol1: -2630.000000, Actual_Vol2: 2698.000000, Actual_Vol3: -2267.000000, Actual_Vol4: 2082.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 692.242497, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.193701, Absolute Angle: 1.608939, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.039607, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1003: 28412 [INFO] CHASSIS_PID_TURN, Time: 11818, Actual_Vol1: -2636.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -2267.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 670.335964, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.190653, Absolute Angle: 1.608886, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.037602

, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
1004: 28412 [INFO] CHASSIS_PID_TURN, Time: 11828, Actual_Vol1: -2735.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -2347.000000, Actual_Vol4: 2156.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 648.406536, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.192943, Absolute Angle: 1.608926, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.033849, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
1005: 28414 [INFO] CHASSIS_PID_TURN, Time: 11838, Actual_Vol1: -2815.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2390.000000, Actual_Vol4: 2236.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 626.468837, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.193770, Absolute Angle: 1.608941, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.029362, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
1006: 28414 [INFO] CHASSIS_PID_TURN, Time: 11848, Actual_Vol1: -2747.000000, Actual_Vol2: 2797.000000, Actual_Vol3: -2402.000000, Actual_Vol4: 2261.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 604.545971, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.192287, Absolute Angle: 1.608915, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026558, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
1007: 28414 [INFO] CHASSIS_PID_TURN, Time: 11858, Actual_Vol1: -2766.000000, Actual_Vol2: 2797.000000, Actual_Vol3: -2427.000000, Actual_Vol4: 2261.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 582.637322, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.190865, Absolute Angle: 1.608890, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026251, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
1008: 28416 [INFO] CHASSIS_PID_TURN, Time: 11868, Actual_Vol1: -2840.000000, Actual_Vol2: 2864.000000, Actual_Vol3: -2421.000000, Actual_Vol4: 2335.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 560.773214, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.186411, Absolute Angle: 1.608812, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024680, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: -0.000000, Actual_Vel3: -15.800000, Actual_Vel4: 117.800000
1009: 28416 [INFO] CHASSIS_PID_TURN, Time: 11878, Actual_Vol1: -2846.000000, Actual_Vol2: 2883.000000, Actual_Vol3: -2396.000000, Actual_Vol4: 2328.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 538.959165, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.181405, Absolute Angle: 1.608725, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027104, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: -0.000000, Actual_Vel3: -15.800000, Actual_Vel4: 117.800000
1010: 28416 [INFO] CHASSIS_PID_TURN, Time: 11888, Actual_Vol1: -2852.000000, Actual_Vol2: 2895.000000, Actual_Vol3: -2396.000000, Actual_Vol4: 2267.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 517.246283, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.171288, Absolute Angle: 1.608548, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.033770, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: 238.000000, Actual_Vel3: -15.800000, Actual_Vel4: 18.000000
1011: 28418 [INFO] CHASSIS_PID_TURN, Time: 11898, Actual_Vol1: -2858.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2384.000000, Actual_Vol4: 2162.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 495.699678, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.154660, Absolute Angle: 1.608258, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.050256, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: 238.000000, Actual_Vel3: -31.600000, Actual_Vel4: 19.200000
1012: 28418 [INFO] CHASSIS_PID_TURN, Time: 11908, Actual_Vol1: -2846.000000, Actual_Vol2: 2864.000000, Actual_Vol3: -2378.000000, Actual_Vol4: 2088.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 474.395587, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.130409, Absolute Angle: 1.607835, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074508, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -31.600000, Actual_Vel4: 19.200000
1013: 28418 [INFO] CHASSIS_PID_TURN, Time: 11918, Actual_Vol1: -2840.000000, Actual_Vol2: 2871.000000, Actual_Vol3: -2384.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 453.413222, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 92.098237, Absolute Angle: 1.607273, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.105771, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -16.200000, Actual_Vel4: 16.000000
1014: 28420 [INFO] CHASSIS_PID_TURN, Time: 11928, Actual_Vol1: -2846.000000, Actual_Vol2: 2803.000000, Actual_Vol3: -2378.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 434.834477, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.857875, Absolute Angle: 1.603078, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.346133, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -16.200000, Actual_Vel4: 14.000000
1015: 28420 [INFO] CHASSIS_PID_TURN, Time: 11938, Actual_Vol1: -2846.000000, Actual_Vol2: 2858.000000, Actual_Vol3: -2372.000000, Actual_Vol4: 2064.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 416.784222, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.805025, Absolute Angle: 1.602156, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.398982, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -16.200000, Actual_Vel4: 14.000000
1016: 28420 [INFO] CHASSIS_PID_TURN, Time: 11948, Actual_Vol1: -2852.000000, Actual_Vol2: 2877.000000, Actual_Vol3: -2328.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 399.227879, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.755634, Absolute Angle: 1.601294, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.448373, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -12.200000, Actual_Vel4: 13.400000
1017: 28422 [INFO] CHASSIS_PID_TURN, Time: 11958, Actual_Vol1: -2852.000000, Actual_Vol2: 2889.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 2027.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 382.180594, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.704728, Absolute Angle: 1.600405, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.495404, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 15.600000, Actual_Vel3: -12.200000, Actual_Vel4: 13.600000
1018: 28422 [INFO] CHASSIS_PID_TURN, Time: 11968, Actual_Vol1: -2852.000000, Actual_Vol2: 2809.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1934.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 365.587579, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.659302, Absolute Angle: 1.599612, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.536119, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 15.600000, Actual_Vel3: -11.200000, Actual_Vel4: 13.600000
1019: 28422 [INFO] CHASSIS_PID_TURN, Time: 11978, Actual_Vol1: -2790.000000, Actual_Vol2: 2858.000000, Actual_Vol3: -2261.000000, Actual_Vol4: 1934.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 349.430636, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.615694, Absolute Angle: 1.598851, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.579159, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 15.600000, Actual_Vel3: -11.200000, Actual_Vel4: 13.600000
1020: 28424 [INFO] CHASSIS_PID_TURN, Time: 11988, Actual_Vol1: -2772.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2267.000000, Actual_Vol4: 1891.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 334.738453, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.469218, Absolute Angle: 1.596295, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.724726, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 15.600000, Actual_Vel3: -11.200000, Actual_Vel4: 12.400000
1021: 28424 [INFO] CHASSIS_PID_TURN, Time: 11998, Actual_Vol1: -2710.000000, Actual_Vol2: 2803.000000, Actual_Vol3: -2168.000000, Actual_Vol4: 1712.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 320.515438, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.422302, Absolute Angle: 1.595476, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.771468, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 15.600000, Actual_Vel3: -11.000000, Actual_Vel4: 12.400000
1022: 28424 [INFO] CHASSIS_PID_TURN, Time: 12008, Actual_Vol1: -776.000000, Actual_Vol2: 715.000000, Actual_Vol3: -499.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 306.804051, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.371139, Absolute Angle: 1.594583, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.822631, over slew: 0, Actual_Vel1: -17.800000, Actual_Vel2: 25.400000, Actual_Vel3: -6.400000, Actual_Vel4: 12.400000
1023: 28426 [INFO] CHASSIS_PID_TURN, Time: 12018, Actual_Vol1: -179.000000, Actual_Vol2: 216.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 294.551132, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.225292, Absolute Angle: 1.592037, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.968478, over slew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: -0.000000, Actual_Vel4: 12.400000
1024: 28426 [INFO] CHASSIS_PID_TURN, Time: 12028, Actual_Vol1: -271.000000, Actual_Vol2: 246.000000, Actual_Vol3: -99.000000, Actual_Vol4: 18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 282.604818, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.194631, Absolute Angle: 1.591502, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.999138, over slew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: -0.000000, Actual_Vel4: 8.400000
1025: 28426 [INFO] CHASSIS_PID_TURN, Time: 12038, Actual_Vol1: -413.000000, Actual_Vol2: 320.000000, Actual_Vol3: -105.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 270.726472, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.187835, Absolute Angle: 1.591384, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.004452, over slew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: -0.000000, Actual_Vel4: 8.400000
1026: 28428 [INFO] CHASSIS_PID_TURN, Time: 12048, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -105.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 258.758599, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.196787, Absolute Angle: 1.591540, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.003030, over slew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: -0.000000, Actual_Vel4: 8.400000
1027: 28428 [INFO] CHASSIS_PID_TURN, Time: 12058, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 246.557861, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.220074, Absolute Angle: 1.591946, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.998576, over slew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: 8.000000, Actual_Vel4: 8.400000
1028: 28428 [INFO] CHASSIS_PID_TURN, Time: 12068, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 234.049552, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.250831, Absolute Angle: 1.592483, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.993570, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 26.600000, Actual_Vel3: 8.000000, Actual_Vel4: 8.400000
1029: 28430 [INFO] CHASSIS_PID_TURN, Time: 12078, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -99.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 221.312258, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.273729, Absolute Angle: 1.592883, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.983454, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 26.600000, Actual_Vel3: 8.000000, Actual_Vel4: 5.200000
1030: 28430 [INFO] CHASSIS_PID_TURN, Time: 12088, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 208.452981, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.285928, Absolute Angle: 1.593096, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.966826, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 26.600000, Actual_Vel3: 16.800000, Actual_Vel4: 5.200000
1031: 28430 [INFO] CHASSIS_PID_TURN, Time: 12098, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 195.638810, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.281417, Absolute Angle: 1.593017, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.942575, over slew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 16.800000, Actual_Vel4: 5.200000
1032: 28432 [INFO] CHASSIS_PID_TURN, Time: 12108, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 182.832356, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.280645, Absolute Angle: 1.593004, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.910402, over slew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 16.800000, Actual_Vel4: 0.000000
1033: 28432 [INFO] CHASSIS_PID_TURN, Time: 12118, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -105.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 170.062333, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.277002, Absolute Angle: 1.592940, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.670040, over slew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 20.600000, Actual_Vel4: 0.000000
1034: 28432 [INFO] CHASSIS_PID_TURN, Time: 12128, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 157.370291, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.269204, Absolute Angle: 1.592804, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.617191, over slew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 20.600000, Actual_Vel4: 0.000000
1035: 28434 [INFO] CHASSIS_PID_TURN, Time: 12138, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 144.751673, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.261862, Absolute Angle: 1.592676, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.567800, over slew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 20.600000, Actual_Vel4: -39.200000
1036: 28434 [INFO] CHASSIS_PID_TURN, Time: 12148, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -105.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 132.181168, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.257051, Absolute Angle: 1.592592, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.516894, over slew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -39.200000
1037: 28434 [INFO] CHASSIS_PID_TURN, Time: 12158, Actual_Vol1: -493.000000, Actual_Vol2: 400.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 119.576957, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.260421, Absolute Angle: 1.592651, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.471467, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -39.200000
1038: 28436 [INFO] CHASSIS_PID_TURN, Time: 12168, Actual_Vol1: -511.000000, Actual_Vol2: 444.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 106.943315, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.263364, Absolute Angle: 1.592702, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.427860, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
1039: 28436 [INFO] CHASSIS_PID_TURN, Time: 12178, Actual_Vol1: -511.000000, Actual_Vol2: 450.000000, Actual_Vol3: -105.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 94.278076, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.266524, Absolute Angle: 1.592757, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.281384, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
1040: 28436 [INFO] CHASSIS_PID_TURN, Time: 12188, Actual_Vol1: -517.000000, Actual_Vol2: 450.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 81.615794, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.266228, Absolute Angle: 1.592752, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.234467, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
1041: 28438 [INFO] CHASSIS_PID_TURN, Time: 12198, Actual_Vol1: -517.000000, Actual_Vol2: 450.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 68.985541, kD: 35.000000, kI: 0.00

0700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.263025, Absolute Angle: 1.592696, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.183304, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1042: 28438 [INFO] CHASSIS_PID_TURN, Time: 12208, Actual_Vol1: -517.000000, Actual_Vol2: 450.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 56.325654, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.265989, Absolute Angle: 1.592748, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098093, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1043: 28438 [INFO] CHASSIS_PID_TURN, Time: 12218, Actual_Vol1: -579.000000, Actual_Vol2: 493.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43.682894, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.264276, Absolute Angle: 1.592718, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098093, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1044: 28440 [INFO] CHASSIS_PID_TURN, Time: 12228, Actual_Vol1: -610.000000, Actual_Vol2: 530.000000, Actual_Vol3: -197.000000, Actual_Vol4: 191.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31.063019, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.261987, Absolute Angle: 1.592678, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098093, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1045: 28440 [INFO] CHASSIS_PID_TURN, Time: 12238, Actual_Vol1: -604.000000, Actual_Vol2: 536.000000, Actual_Vol3: -216.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18.443358, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.261966, Absolute Angle: 1.592678, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.089140, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1046: 28440 [INFO] CHASSIS_PID_TURN, Time: 12248, Actual_Vol1: -604.000000, Actual_Vol2: 530.000000, Actual_Vol3: -228.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5.796962, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.264640, Absolute Angle: 1.592724, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.065854, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1047: 28442 [INFO] CHASSIS_PID_TURN, Time: 12258, Actual_Vol1: -604.000000, Actual_Vol2: 536.000000, Actual_Vol3: -222.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -6.812050, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.260901, Absolute Angle: 1.592659, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035097, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1048: 28442 [INFO] CHASSIS_PID_TURN, Time: 12268, Actual_Vol1: -616.000000, Actual_Vol2: 536.000000, Actual_Vol3: -234.000000, Actual_Vol4: 216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -19.411581, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.259953, Absolute Angle: 1.592642, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028877, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1049: 28442 [INFO] CHASSIS_PID_TURN, Time: 12278, Actual_Vol1: -616.000000, Actual_Vol2: 579.000000, Actual_Vol3: -228.000000, Actual_Vol4: 216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -32.036665, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.262508, Absolute Angle: 1.592687, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028877, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1050: 28444 [INFO] CHASSIS_PID_TURN, Time: 12288, Actual_Vol1: -684.000000, Actual_Vol2: 628.000000, Actual_Vol3: -296.000000, Actual_Vol4: 290.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -44.668432, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.263177, Absolute Angle: 1.592699, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024367, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1051: 28444 [INFO] CHASSIS_PID_TURN, Time: 12298, Actual_Vol1: -696.000000, Actual_Vol2: 634.000000, Actual_Vol3: -302.000000, Actual_Vol4: 302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -57.293692, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.262526, Absolute Angle: 1.592687, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023595, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1052: 28444 [INFO] CHASSIS_PID_TURN, Time: 12308, Actual_Vol1: -696.000000, Actual_Vol2: 634.000000, Actual_Vol3: -302.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -69.955795, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.266210, Absolute Angle: 1.592752, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1053: 28446 [INFO] CHASSIS_PID_TURN, Time: 12318, Actual_Vol1: -696.000000, Actual_Vol2: 628.000000, Actual_Vol3: -308.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -82.603629, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.264783, Absolute Angle: 1.592727, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012154, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1054: 28446 [INFO] CHASSIS_PID_TURN, Time: 12328, Actual_Vol1: -702.000000, Actual_Vol2: 628.000000, Actual_Vol3: -308.000000, Actual_Vol4: 302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -95.261112, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.265748, Absolute Angle: 1.592744, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.009473, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1055: 28446 [INFO] CHASSIS_PID_TURN, Time: 12338, Actual_Vol1: -708.000000, Actual_Vol2: 634.000000, Actual_Vol3: -308.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -107.916924, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.265581, Absolute Angle: 1.592741, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.009473, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1056: 28448 [INFO] CHASSIS_PID_TURN, Time: 12348, Actual_Vol1: -813.000000, Actual_Vol2: 671.000000, Actual_Vol3: -308.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -120.568364, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.265144, Absolute Angle: 1.592733, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.006571, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000

1057: WARNING - pros.serial.terminal.terminal:stop - Stopping terminal

```
1:  /**
2:   * @file: ./RobotCode/include/fonts/fonts.h
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/16/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see: ../src/objects/lcdCode/fonts/
8:   *
9:   * contains definitions for fonts to use for lvgl
10:  * this is used to make gui more interesting and provide more contrast and have
11:  * the ability to fit more data because of a smaller font
12:  */
13:
14:  #ifndef __FONTS_H__
15:  #define __FONTS_H__
16:
17:
18:  #ifdef USE_DEJAVU_9
19:  LV_FONT_DECLARE(dejavu_9);
20:  extern lv_font_t dejavu_9;
21:  #endif
22:
23:  #ifdef USE_DEJAVU_12
24:  LV_FONT_DECLARE(dejavu_12);
25:  extern lv_font_t dejavu_12;
26:  #endif
27:
28:  #ifdef USE_DEJAVU_16
29:  LV_FONT_DECLARE(dejavu_16);
30:  extern lv_font_t dejavu_16;
31:  #endif
32:
33:  #endif
```

```cpp
#include <csignal>

#include "main.h"

#include "Autons.hpp"
#include "Configuration.hpp"
#include "DriverControl.hpp"
#include "objects/controller/controller.hpp"
#include "objects/lcdCode/DriverControl/AutonomousLCD.hpp"
#include "objects/lcdCode/DriverControl/DriverControlLCD.hpp"
#include "objects/lcdCode/gui.hpp"
#include "objects/lcdCode/TemporaryScreen.hpp"
#include "objects/motors/Motors.hpp"
#include "objects/motors/MotorThread.hpp"
#include "objects/position_tracking/PositionTracker.hpp"
#include "objects/serial/Logger.hpp"
#include "objects/serial/Server.hpp"
#include "objects/subsystems/chassis.hpp"

int final_auton_choice;
AutonomousLCD auton_lcd;

/**
 * Runs initialization code. This occurs as soon as the program is started.
 *
 * All other competition modes are blocked by initialize; it is recommended
 * to keep execution time for this mode under a few seconds.
 */
void initialize()
{
    pros::c::serctl(SERCTL_ACTIVATE, 0);  // I think this enables stdin (necessary to start server)

    Motors::register_motors();
    MotorThread::get_instance()->start_thread();

    pros::delay(100); //wait for terminal to start and lvgl
    Configuration* config = Configuration::get_instance();
    config->init();
    config->print_config_options();

    final_auton_choice = chooseAuton();
    Autons auton;
    config->filter_color = auton.AUTONOMOUS_COLORS.at(final_auton_choice);
    auton.set_autonomous_number(final_auton_choice);

    Sensors::calibrate_imu();


    // std::cout << OptionsScreen::cnfg.use_hardcoded << '\n';
    // std::cout << OptionsScreen::cnfg.gyro_turn << '\n';
    // std::cout << OptionsScreen::cnfg.accelleration_ctrl << '\n';
    // std::cout << OptionsScreen::cnfg.check_motor_tmp << '\n';
    // std::cout << OptionsScreen::cnfg.use_previous_macros << '\n';
    // std::cout << OptionsScreen::cnfg.record << '\n';

    std::cout << "initalize finished" << "\n";
    lv_scr_load(tempScreen::temp_screen);
}



/**
 * Runs while the robot is in the disabled state of Field Management System or
 * the VEX Competition Switch, following either autonomous or opcontrol. When
 * the robot is enabled, this task will exit.
 */
void disabled() {}



/**
 * Runs after initialize(), and before autonomous when connected to the Field
 * Management System or the VEX Competition Switch. This is intended for
 * competition-specific initialization routines, such as an autonomous selector
 * on the LCD.
 *
 * This task will exit when the robot is enabled and autonomous or opcontrol
 * starts.
 */
void competition_initialize() {
    Autons auton;
    auton_lcd.update_labels(auton.get_autonomous_number());
}



/**
 * Runs the user autonomous code. This function will be started in its own task
 * with the default priority and stack size whenever the robot is enabled via
 * the Field Management System or the VEX Competition Switch in the autonomous
 * mode. Alternatively, this function may be called in initialize or opcontrol
 * for non-competition testing purposes.
 *
 * If the robot is disabled or communications is lost, the autonomous task
 * will be stopped. Re-enabling the robot will restart the task, not re-start it
 * from where it left off.
 */
void autonomous() {
    Autons auton;
    auton.run_autonomous();
}


void log_thread_fn( void* )
{
    // TODO: add back imu functionality when imu is mounted
    Logger logger;
    Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);

    Configuration* config = Configuration::get_instance();
    double kP = config->chassis_pid.kP;
    double kI = config->chassis_pid.kI;
    double kD = config->chassis_pid.kD;
```

```
114:     double I_max = config->chassis_pid.I_max;
115:
116:     int l_id = Sensors::left_encoder.get_unique_id();
117:     int r_id = Sensors::right_encoder.get_unique_id();
118:     // double prev_l_encoder = std::get<0>(chassis.get_average_encoders(l_id, r_id));
119:     // double prev_r_encoder = std::get<1>(chassis.get_average_encoders(l_id, r_id));
120:     // double intitial_angle = Sensors::imu.get_heading();
121:     // double prev_angle = Sensors::imu.get_heading();
122:     // double relative_angle = 0;
123:     //
124:     while ( 1 )
125:     {
126:         // double delta_l = std::get<0>(chassis.get_average_encoders(l_id, r_id)) - prev_l_encoder;
127:         // double delta_r = std::get<1>(chassis.get_average_encoders(l_id, r_id)) - prev_r_encoder;
128:         // double delta_theta = chassis.calc_delta_theta(prev_angle, delta_l, delta_r);
129:         // prev_angle = prev_angle + delta_theta;
130:         // relative_angle = relative_angle + delta_theta;
131:         //
132:
133:         std::string msg = (
134:             "[INFO] " + std::string("CHASSIS_PID ")
135:             + ", Actual_Vol1: " + std::to_string(Motors::front_left.get_actual_voltage())
136:             + ", Actual_Vol2: " + std::to_string(Motors::front_right.get_actual_voltage())
137:             + ", Actual_Vol3: " + std::to_string(Motors::back_left.get_actual_voltage())
138:             + ", Actual_Vol4: " + std::to_string(Motors::back_right.get_actual_voltage())
139:             + ", Slew: " + std::to_string(0)
140:             + ", Brake: " + std::to_string(Motors::front_left.get_brake_mode())
141:             + ", Gear: " + std::to_string(Motors::front_left.get_gearset())
142:             + ", I_max: " + std::to_string(I_max)
143:             + ", I: " + std::to_string(0)
144:             + ", kD: " + std::to_string(kD)
145:             + ", kI: " + std::to_string(kI)
146:             + ", kP: " + std::to_string(kP)
147:             + ", Time: " + std::to_string(pros::millis())
148:             + ", Position_Sp: " + std::to_string(1269.32)
149:             + ", position_l: " + std::to_string(Sensors::left_encoder.get_position(l_id))
150:             + ", position_r: " + std::to_string(Sensors::right_encoder.get_position(r_id))
151:             + ", Heading_Sp: " + std::to_string(0)
152:             + ", Relative_Heading: " + std::to_string(0)
153:             + ", Actual_Vel1: " + std::to_string(Motors::front_left.get_actual_velocity())
154:             + ", Actual_Vel2: " + std::to_string(Motors::front_right.get_actual_velocity())
155:             + ", Actual_Vel3: " + std::to_string(Motors::back_left.get_actual_velocity())
156:             + ", Actual_Vel4: " + std::to_string(Motors::back_right.get_actual_velocity())
157:         );
158:         // std::cout << msg << "\n";
159:         pros::delay(10);
160:     }
161: }
162:
163:
164:
165: void Exit( int signal )
166: {
167:     //Writer writer;
168:     std::cerr << "program caught " << signal << "\n" << std::flush;
169:     std::cerr << "errno: " << errno << "\n" << std::flush;
170:     std::cerr << "strerror: " << std::strerror(errno) << "\n" << std::flush;
171:     pros::delay(100); // wait for stdout to be flushed
172:     raise(signal);
173: }
174:
175:
176:
177: /**
178:  * Runs the operator control code. This function will be started in its own task
179:  * with the default priority and stack size whenever the robot is enabled via
180:  * the Field Management System or the VEX Competition Switch in the operator
181:  * control mode.
182:  *
183:  * If no competition control is connected, this function will run immediately
184:  * following initialize().
185:  *
186:  * If the robot is disabled or communications is lost, the
187:  * operator control task will be stopped. Re-enabling the robot will restart the
188:  * task, not resume it from where it left off.
189:  */
190: void opcontrol() {
191:     // pros::ADIAnalogIn l1 (1);
192:     // pros::ADIAnalogIn l2 (2);
193:     // pros::ADIAnalogIn l3 (3);
194:     // while(1) {
195:     //     std::cout << l1.get_value() << " " << l2.get_value() << " " << l3.get_value() << "\n";
196:     //     pros::delay(50);
197:     // }
198:     // Logger logger;
199:     // pros::ADIDigitalIn limit_switch('A');
200:
201:
202:     // pros::Task write_task (log_thread_fn,
203:     //                        (void*)NULL,
204:     //                        TASK_PRIORITY_DEFAULT,
205:     //                        TASK_STACK_DEPTH_DEFAULT,
206:     //                        "logger_thread");
207:
208:
209:
210:     Server server;
211:     server.clear_stdin();
212:     server.start_server();
213:     server.set_debug_mode(true);
214:
215:     // int stop = pros::millis() + 8000;
216:     //
217:     // Lift lift(Motors::lift, {0, 800});
218:     // while ( pros::millis() < stop )
219:     // {
220:     //     lift.move_to(900, false, true);
221:     //     pros::delay(10);
222:     // }
223:     // stop = pros::millis() + 2000;
224:     // while ( pros::millis() < stop )
225:     // {
226:     //     lift.move_to(0, false, true);
```

```
227:     //     pros::delay(10);
228:     // }
229:     // logger.dump();
230:     // logger.dump();
231:     // logger.dump();
232:     // logger.dump();
233:     // logger.dump();
234:     // logger.dump();
235:     // logger.dump();
236:     // logger.dump();
237:     //
238:     // Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, 12.4 );
239:     // int stop = pros::millis() + 8000;
240:     //
241:     // chassis.turn_left(13, 12000, INT32_MAX, true, false, true);
242:     //
243:     // while ( pros::millis() < stop )
244:     // {
245:     //     chassis.turn_left(13, 12000, INT32_MAX, false, false, true );
246:     //     pros::delay(10);
247:     // }
248:
249:     std::cout << "opcontrol started\n";
250:
251:     std::signal(SIGSEGV, Exit);
252:     std::signal(SIGTERM, Exit);
253:     std::signal(SIGINT, Exit);
254:     std::signal(SIGILL, Exit);
255:     std::signal(SIGABRT, Exit);
256:     std::signal(SIGFPE, Exit);
257:     std::signal(SIGBUS, Exit);
258:     std::signal(SIGALRM, Exit);
259:     std::signal(SIGSTOP, Exit);
260:     std::signal(SIGUSR1, Exit);
261:     std::signal(SIGUSR2, Exit);
262:     std::signal(SIGKILL, Exit);
263:
264:     pros::delay(100);
265:
266:     lv_scr_load(tempScreen::temp_screen);
267:
268:
269:     Controller controllers;
270:     // while(1) {
271:     //     Motors::left_intake.user_move(controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y));
272:     //     Motors::right_intake.user_move(controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y));
273:     //     std::cout << Motors::left_intake.get_torque() << " " << Motors::left_intake.get_efficiency() << " " << Motors::left_intake.get_actual_voltage() << "\n";
274:     //     std::cout << Motors::right_intake.get_torque() << " " << Motors::right_intake.get_efficiency() << " " << Motors::right_intake.get_actual_voltage() << "\n";
275:     //     std::cout << "\n";
276:     //     pros::delay(10);
277:     // }
278:
279:
280:     // Motors motors;
281:     // Motors::record_macro();
282:     //
283:     // Writer writer;
284:     // while( writer.get_count() > 0 )
285:     // {
286:     //     std::cout << pros::millis() << " " << writer.get_count() << "\n";
287:     //     pros::delay(1);
288:     // }
289:     //
290:     // std::cout << "done\n";
291:
292:     //update controller with color of cube and if it is loaded or not
293:
294:     // Controller controllers;
295:     // std::string controller_text = "no cube loaded";
296:     // std::string prev_controller_text = "";
297:
298:     // PositionTracker* tracker = PositionTracker::get_instance();
299:     // tracker->start_thread();
300:     // std::cout << pros::Task::get_count() << "\n";
301:     // Chassis chassis(Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, Sensors::imu, 12.75, 5/3, 3.25);
302:     DriverControlLCD lcd;
303:     // lcd.update_labels();
304:     // chassis.generate_profiles();
305:
306:
307:     // chassis.turn_left(90);  // passing
308:     // pros::delay(1000);
309:     //
310:     // chassis.turn_right(90); // passing
311:     // pros::delay(1000);
312:     //
313:     // chassis.turn_left(45); // passing
314:     // pros::delay(1000);
315:     //
316:     // chassis.turn_right(45); // passing
317:     // pros::delay(1000);
318:     //
319:     // chassis.turn_left(30); // passing
320:     // pros::delay(1000);
321:     //
322:     // chassis.turn_right(30); // passing
323:     // pros::delay(1000);
324:     //
325:     // chassis.turn_left(10); // passing
326:     // pros::delay(1000);
327:     //
328:     // chassis.turn_right(10); // passing
329:     // pros::delay(1000);
330:
331:     // chassis.turn_right(270); // passing
332:     // pros::delay(1000);
333:     //
334:     // chassis.turn_left(270); // passing
335:     // pros::delay(1000);
336:     //
337:
338:     // chassis.turn_to_angle(90);  // passing
339:     // pros::delay(1000);
```

```
340:     //
341:     // chassis.turn_to_angle(270);  // passing
342:     // pros::delay(1000);
343:     //
344:     // chassis.turn_to_angle(0);  // passing
345:     // pros::delay(1000);
346:     //
347:     // chassis.turn_to_angle(-90);  // passing
348:     // pros::delay(1000);
349:
350:     // chassis.turn_to_point(36, 0);  // passing
351:     // pros::delay(1000);
352:     // chassis.turn_to_point(-36, 0);  // passing
353:     // pros::delay(1000);
354:     // chassis.turn_to_point(0, 36);  // passing
355:     // pros::delay(1000);
356:     // chassis.turn_to_point(-36, -36);  // passing
357:
358:     // chassis.drive_to_point(0, 36);  // passing
359:     // pros::delay(1000);
360:     // chassis.drive_to_point(36, 36);  // passing
361:     // pros::delay(1000);
362:     // chassis.drive_to_point(36, 0);  // passing
363:     // pros::delay(1000);
364:     // chassis.drive_to_point(0, 0);  // passing
365:
366:     // tracker->stop_logging();
367:     lcd.update_labels();
368:     Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
369:     PositionTracker* tracker = PositionTracker::get_instance();
370:     tracker->enable_imu();
371:     tracker->start_thread();
372:     // chassis.turn_left(90);
373:     Autons autons;
374:     autons.skills2();
375:
376:     // gather data from position tracker
377:     // tracker->start_logging();
378:     // while(1) {
379:     //     pros::delay(10);
380:     // }
381:
382:     pros::Task driver_control_task (driver_control,
383:                                     (void*)NULL,
384:                                     TASK_PRIORITY_DEFAULT,
385:                                     TASK_STACK_DEPTH_DEFAULT,
386:                                     "DriverControlTask");
387:
388:
389:
390:     // double prev_angle = std::fmod(Sensors::imu.get_heading() + 360, 360);
391:     // double ref_angle = std::fmod(Sensors::imu.get_heading() + 360, 360);
392:     int l_id = Sensors::left_encoder.get_unique_id();
393:     int r_id = Sensors::right_encoder.get_unique_id();
394:     int s_id = Sensors::strafe_encoder.get_unique_id();
395:     // double prev_l = Sensors::left_encoder.get_position(l_id);
396:     // double prev_r = Sensors::right_encoder.get_position(r_id);
397:     // pros::delay(1);
398:     // chassis.straight_drive(-1000, 0, 12000, 10000);
399:     // Sensors::ball_detector.start_logging();
400:     // Logger::stop_queueing();
401:     while(1)
402:     {
403:         // print encoder values
404:
405:         // std::cout << "r: " << Sensors::right_encoder.get_position(r_id) << " | l: " << Sensors::left_encoder.get_position(l_id) << " | s: " << Sensors::strafe_encoder.get_position(s_id) << "\n";
406:         // double delta_theta = chassis.calc_delta_theta(prev_angle, ref_angle, Sensors::left_encoder.get_position(l_id) - prev_l, Sensors::right_encoder.get_position(r_id) - prev_r);
407:         // prev_angle = prev_angle + delta_theta;
408:         // prev_l = Sensors::left_encoder.get_position(l_id);
409:         // prev_r = Sensors::right_encoder.get_position(r_id);
410:         //
411:         // std::cout << "delta theta: " << delta_theta << "  |  new angle: " << prev_angle << "\n";
412:         // std::cout << tracker->get_position().x_pos << " " << tracker->get_position().y_pos << " " << tracker->to_degrees(tracker->get_position().theta) << "\n";
413:         lcd.update_labels();
414:         // server.handle_requests(50);
415:         // std::cout << "handling requests\n";
416:         // logger.dump();
417:
418:         pros::delay(20);
419:     }
420: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/Autons.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 12/5/19
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains class that holds data about the autonomous period as well as
8:   * structs for configuration data
9:   */
10:
11: #ifndef __AUTONS_HPP__
12: #define __AUTONS_HPP__
13:
14: #include <unordered_map>
15:
16: #include "main.h"
17:
18:
19: typedef struct
20: {
21:     bool use_hardcoded = 0;
22:     bool gyro_turn = 1;
23:     bool accelleration_ctrl = 1;
24:     bool check_motor_tmp = 0;
25:     bool use_previous_macros = 1;
26:     bool record = 0;
27: } autonConfig;
28:
29:
30:
31: /**
32:  * @see: Motors.hpp
33:  * @see: ./objects/lcdCode
34:  *
35:  * contains data for the autonomous period as well as functions to run the
36:  * selected autonomous
37:  */
38: class Autons
39: {
40:     private:
41:         static int selected_number;
42:
43:     public:
44:         Autons();
45:         ~Autons();
46:
47:
48:         int debug_auton_num;      //change if more autons are added
49:                                   //debugger should be last option
50:         int driver_control_num;
51:
52:         const std::unordered_map <int, const char*> AUTONOMOUS_NAMES = {
53:             {1, "Driver Control"},          //used to find name of auton
54:             {2, "one_pt"},                  //to keep title the same
55:             {3, "skills-47"},
56:             {4, "skills-66"},
57:             {5, "blue north"},
58:             {6, "blue north 2"},
59:             {7, "red north"},
60:             {8, "Debugger"}
61:         };
62:         const std::unordered_map <int, const char*> AUTONOMOUS_DESCRIPTIONS = {   //used to find color of auton
63:             {1, "goes directly to\ndriver control"},                          //selected to keep background the same
64:             {2, "drives forward and\nbackwards"},
65:             {3, "skills auton that scores 47 points"},
66:             {4, "skills auton that scores 66 points"},
67:             {5, "Goes to cap middle wall\ntower and then cycle\nstarting tower"},
68:             {6, "cycles closest tower, turns left"},
69:             {7, "cylces closest tower, turns right"},
70:             {8, "opens debugger"}
71:         };
72:         const std::unordered_map <int, std::string> AUTONOMOUS_COLORS = {
73:             {1, "none"},                  //used to find color of auton
74:             {2, "none"},                  //selected to keep background the same
75:             {3, "none"},
76:             {4, "none"},
77:             {5, "blue"},
78:             {6, "blue"},
79:             {7, "red"},
80:             {8, "none"}
81:         };
82:
83:         void set_autonomous_number(int n);
84:         int get_autonomous_number();
85:
86:
87:         /**
88:          * @return: None
89:          *
90:          * @see: Motors.hpp
91:          * @see: Chassis.hpp
92:          *
93:          * get robot ready for auton
94:          */
95:         void deploy();
96:
97:         /**
98:          * @param: autonConfig cnfg -> the configuration to use for the auton
99:          * @return: None
100:         *
101:         * @see: Motors.hpp
102:         *
103:         * drives forward
104:         */
105:        void one_pt();
106:
107:        /**
108:         * @param: autonConfig cnfg -> the configuration to use for the auton
109:         * @return: None
110:         *
111:         * @see: Motors.hpp
112:         *
113:         * runs skills
```

```
114:         */
115:        void skills();
116:
117:        void skills2();
118:
119:
120:        void blue_north();
121:        void blue_north_2();
122:        void red_north();
123:
124:        void run_autonomous();
125:    };
126:
127:
128:
129:
130:    #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/Autons.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 12/5/19
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see: Autons.hpp
8:   *
9:   * contains implementation for autonomous options
10:  */
11:
12:  #include <unordered_map>
13:
14:  #include "main.h"
15:
16:  #include "Autons.hpp"
17:  #include "objects/motors/Motors.hpp"
18:  #include "objects/motors/MotorThread.hpp"
19:  #include "objects/position_tracking/PositionTracker.hpp"
20:  #include "objects/subsystems/chassis.hpp"
21:  #include "objects/subsystems/Indexer.hpp"
22:  #include "objects/subsystems/intakes.hpp"
23:  #include "objects/lcdCode/DriverControl/AutonomousLCD.hpp"
24:
25:
26:  int Autons::selected_number = 1;
27:
28:  Autons::Autons( )
29:  {
30:      debug_auton_num = 8;
31:      driver_control_num = 1;
32:  }
33:
34:
35:
36:  Autons::~Autons( ) {
37:
38:  }
39:
40:  void Autons::set_autonomous_number(int n) {
41:      selected_number = n;
42:  }
43:
44:  int Autons::get_autonomous_number() {
45:      return selected_number;
46:  }
47:
48:  /**
49:   * deploys by outtaking and running top roller
50:   */
51:  void Autons::deploy() {
52:      Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "none");
53:      Intakes intakes(Motors::left_intake, Motors::right_intake);
54:
55:      intakes.rocket_outwards();
56:      indexer.run_upper_roller();
57:
58:      pros::delay(1000);
59:
60:      intakes.stop();
61:      indexer.stop();
62:  }
63:
64:
65:  /**
66:   * drives forward to score in the zone, then drive backward
67:   * to stop touching the cube
68:   */
69:  void Autons::one_pt() {
70:
71:      deploy();
72:  }
73:
74:
75:
76:
77:  /**
78:   * runs unit test
79:   * 180 degree, 90 degree, 45 degree, 45 degree
80:   * tilter movement
81:   * straight drive moving
82:   */
83:  void Autons::skills() {
84:      Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
85:      Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
86:      Intakes intakes(Motors::left_intake, Motors::right_intake);
87:      PositionTracker* tracker = PositionTracker::get_instance();
88:      tracker->start_thread();
89:      tracker->enable_imu();
90:      tracker->set_log_level(0);
91:      tracker->set_position({0, 0, 0});
92:
93:      deploy();
94:
95:      // tower one
96:      int uid = chassis.profiled_straight_drive(1000, 450, 3000, true);
97:      while(!chassis.is_finished(uid)) {
98:          intakes.intake();
99:          indexer.auto_increment();
100:         pros::delay(10);
101:     }
102:
103:     intakes.stop();
104:     indexer.stop();
105:
106:     chassis.turn_left(82, 450, 2500);
107:
108:     chassis.profiled_straight_drive(975, 450, 2500);
109:
110:     indexer.index();
111:     pros::delay(300);
112:     indexer.stop();
113:
```

```
114:     // tower two
115:     chassis.profiled_straight_drive(-1000, 450, 2500);
116:     chassis.turn_right(107, 450, 3500);
117:
118:     intakes.hold_outward();
119:     chassis.pid_straight_drive(1225, 0, 450, 3500);
120:
121:     uid = chassis.pid_straight_drive(500, 0, 300, 3000, true);
122:     while(!chassis.is_finished(uid)) {
123:         intakes.intake();
124:         indexer.auto_increment();
125:         pros::delay(10);
126:     }
127:
128:     intakes.stop();
129:     indexer.stop();
130:
131:     chassis.turn_left(62, 450, 2000);
132:
133:     chassis.pid_straight_drive(345, 0, 450, 2000);
134:
135:     indexer.index();
136:     pros::delay(300);
137:     indexer.stop();
138:
139:     pros::delay(500);
140:
141:     indexer.index();
142:     pros::delay(300);
143:     indexer.stop();
144:
145:     // tower 3
146:     chassis.pid_straight_drive(-450, 0, 450, 2000);
147:     chassis.turn_right(78, 450, 2000);
148:
149:     intakes.hold_outward();
150:     chassis.pid_straight_drive(1500, 0, 450, 3500);
151:
152:     uid = chassis.pid_straight_drive(600, 0, 300, 3000, true);
153:     while(!chassis.is_finished(uid)) {
154:         intakes.intake();
155:         indexer.auto_increment();
156:         pros::delay(10);
157:     }
158:
159:
160:     chassis.turn_left(62, 450, 2000);
161:
162:     chassis.pid_straight_drive(1175, 0, 450, 2000);
163:
164:     indexer.index();
165:     pros::delay(300);
166:     indexer.stop();
167:
168:     //
169:     // chassis.drive_to_point(15, 15, 0, 0, 150, INT32_MAX, false);
170:     // chassis.pid_straight_drive(400);
171: }
172:
173: void Autons::skills2() {
174:     Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
175:     Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
176:     Intakes intakes(Motors::left_intake, Motors::right_intake);
177:     PositionTracker* tracker = PositionTracker::get_instance();
178:     tracker->start_thread();
179:     tracker->enable_imu();
180:     tracker->set_log_level(0);
181:     tracker->set_position({0, 0, 0});
182:
183:     deploy();
184:
185:     // tower one
186:     int uid = chassis.profiled_straight_drive(1000, 450, 3000, true);
187:     while(!chassis.is_finished(uid)) {
188:         intakes.intake();
189:         indexer.auto_increment();
190:         pros::delay(10);
191:     }
192:
193:     intakes.stop();
194:     indexer.stop();
195:
196:     chassis.turn_left(82, 450, 2500);
197:
198:     chassis.profiled_straight_drive(975, 450, 2500);
199:
200:     indexer.index();
201:     pros::delay(300);
202:     indexer.stop();
203:
204:     // tower two
205:     chassis.profiled_straight_drive(-1000, 450, 2500);
206:     chassis.turn_right(107, 450, 3500);
207:
208:     uid = chassis.pid_straight_drive(2000, 0, 450, 5000, true);
209:     while(!chassis.is_finished(uid)) {
210:         intakes.intake();
211:         indexer.auto_increment();
212:         pros::delay(10);
213:     }
214: }
215:
216:
217:
218: void Autons::blue_north() {
219:     Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
220:     Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
221:     indexer.update_filter_color("red");
222:     Intakes intakes(Motors::left_intake, Motors::right_intake);
223:     PositionTracker* tracker = PositionTracker::get_instance();
224:     tracker->start_thread();
225:     tracker->enable_imu();
226:     tracker->set_log_level(0);
```

```
227:     tracker->set_position({0, 0, 0});
228:
229:     deploy();
230:
231:     chassis.drive_to_point(0, 27);
232:
233:     chassis.drive_to_point(27.2, 5.5, 0, 1, 125, INT32_MAX, false);
234:
235:     chassis.turn_right(51);
236:     chassis.pid_straight_drive(235);
237:
238:     // chassis.pid_straight_drive(200, 0, 80, INT32_MAX, false, false);
239:
240:     for(int i=0; i < 50; i++) {  // score preload
241:         indexer.auto_index();
242:         pros::delay(10);
243:     }
244:
245:     chassis.drive_to_point(0, 27);
246:
247:     intakes.hold_outward();
248:
249:     chassis.drive_to_point(-17.7, 9.7, 0, 1, 100, INT32_MAX, false);
250:     // chassis.pid_straight_drive(1100, 0, 150, INT32_MAX, false, false);
251:
252:     int uid = chassis.pid_straight_drive(450, 0, 80, 2000, true, false);
253:     while(!chassis.is_finished(uid)) {
254:         indexer.auto_increment();
255:         intakes.intake();
256:     }
257:
258:     // intakes.intake();
259:     // indexer.index_until_filtered();
260:     // pros::delay(100);
261:     // intakes.stop();
262:     //
263:     for(int i=0; i < 100; i++) {  // index a little bit longer
264:         indexer.auto_index();
265:         pros::delay(10);
266:     }
267:
268:     chassis.pid_straight_drive(-500, 0, 200, INT32_MAX, false, false);
269: }
270:
271:
272: void Autons::blue_north_2() {
273:     Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
274:     Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
275:     indexer.update_filter_color("blue");
276:     Intakes intakes(Motors::left_intake, Motors::right_intake);
277:     PositionTracker* tracker = PositionTracker::get_instance();
278:     tracker->start_thread();
279:     tracker->enable_imu();
280:     tracker->set_log_level(0);
281:     tracker->set_position({0, 0, 0});
282:
283:     deploy();
284:
285:     chassis.pid_straight_drive(560, 0, 100);
286:     chassis.turn_left(136, 100, 5000);
287:
288:     intakes.hold_outward();
289:     // chassis.pid_straight_drive(300, 0, 80, 2000, false, true);
290:
291:
292:     intakes.intake();
293:     indexer.auto_increment();
294:     chassis.pid_straight_drive(900, 0, 80, 2000, false, false);
295:     indexer.stop();
296:     // indexer.index_until_filtered();
297:     intakes.stop();
298:
299:     for(int i=0; i < 150; i++) {  // index a little bit longer
300:         // intakes.intake();
301:         indexer.index_no_backboard();
302:         pros::delay(10);
303:     }
304:
305:     //
306:     // for(int i=0; i < 100; i++) {  // index a little bit longer
307:     //     // intakes.intake();
308:     //     indexer.index();
309:     //     pros::delay(10);
310:     // }
311:
312:     indexer.fix_ball();
313:     pros::delay(1000);
314:     indexer.stop();
315:     for(int i=0; i < 40; i++) {  // index a little bit longer
316:         intakes.intake();
317:         // indexer.auto_index();
318:         pros::delay(10);
319:     }
320:     intakes.stop();
321:
322:     intakes.hold_outward();
323:     chassis.pid_straight_drive(-400, 0, 80);
324: }
325:
326:
327: void Autons::red_north() {
328:     Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
329:     Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
330:     indexer.update_filter_color("blue");
331:     Intakes intakes(Motors::left_intake, Motors::right_intake);
332:     PositionTracker* tracker = PositionTracker::get_instance();
333:     tracker->start_thread();
334:     tracker->enable_imu();
335:     tracker->set_log_level(0);
336:     tracker->set_position({0, 0, 0});
337:
338:     deploy();
339:
```

```
340:     chassis.pid_straight_drive(560, 0, 100);
341:     chassis.turn_right(136, 100, 5000);
342:
343:     intakes.hold_outward();
344:     // chassis.pid_straight_drive(300, 0, 80, 2000, false, true);
345:
346:
347:     intakes.intake();
348:     indexer.auto_increment();
349:     chassis.pid_straight_drive(900, 0, 80, 2000, false, false);
350:     indexer.stop();
351:     // indexer.index_until_filtered();
352:     intakes.stop();
353:
354:     for(int i=0; i < 150; i++) {  // index a little bit longer
355:        // intakes.intake();
356:        indexer.index_no_backboard();
357:        pros::delay(10);
358:     }
359:
360:     //
361:     // for(int i=0; i < 100; i++) {  // index a little bit longer
362:     //    // intakes.intake();
363:     //    indexer.index();
364:     //    pros::delay(10);
365:     // }
366:
367:     indexer.fix_ball();
368:     pros::delay(1000);
369:     indexer.stop();
370:     for(int i=0; i < 40; i++) {  // index a little bit longer
371:        intakes.intake();
372:        // indexer.auto_index();
373:        pros::delay(10);
374:     }
375:     intakes.stop();
376:
377:     intakes.hold_outward();
378:     chassis.pid_straight_drive(-400, 0, 80);
379:  }
380:
381:  void Autons::run_autonomous() {
382:     switch(selected_number)
383:     {
384:        case 1:
385:           break;
386:
387:        case 2:
388:           one_pt();
389:           break;
390:
391:        case 3:
392:           skills();
393:           break;
394:
395:        case 4:
396:           skills2();
397:           break;
398:
399:        case 5:
400:           blue_north();
401:           break;
402:
403:        case 6:
404:           blue_north_2();
405:           break;
406:
407:        case 7:
408:           red_north();
409:     }
410:  }
```

```
1:  /**
2:   * @file: ./RobotCode/src/Configuration.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains class static variables for runtime configuration
8:   *
9:   */
10:
11:  #ifndef __CONFIGURATION_HPP__
12:  #define __CONFIGURATION_HPP__
13:
14:  #include <iostream>
15:  #include <vector>
16:
17:  #include "main.h"
18:
19:  #include "../lib/json.hpp"
20:
21:
22:  #define LEFT_ENC_TOP_PORT       'G'
23:  #define LEFT_ENC_BOTTOM_PORT    'H'
24:  #define RIGHT_ENC_TOP_PORT      'A'
25:  #define RIGHT_ENC_BOTTOM_PORT   'B'
26:  #define STRAFE_ENC_TOP_PORT     'E'
27:  #define STRAFE_ENC_BOTTOM_PORT  'F'
28:  #define DETECTOR_MIDDLE_PORT    'C'
29:  #define POTENTIOMETER_PORT      'Z'
30:
31:  #define DETECTOR_BOTTOM_PORT    'Z'  // no port available but still wanted in code
32:  #define DETECTOR_TOP_PORT       'D'  // no port available but still wanted in code
33:
34:  #define OPTICAL_PORT            5
35:  #define IMU_PORT                10
36:
37:
38:  typedef struct
39:  {
40:      double kP = 0;
41:      double kI = 0;
42:      double kD = 0;
43:      double I_max = 0;
44:      void print() {
45:          std::cout << "kP: " << this->kP << "\n";
46:          std::cout << "kI: " << this->kI << "\n";
47:          std::cout << "kD: " << this->kD << "\n";
48:          std::cout << "I_max: " << this->I_max << "\n";
49:      };
50:  } pid;
51:
52:
53:
54:  /**
55:   * @see: ../lib/json.hpp
56:   *
57:   * Singleton class
58:   * contains class to read data from config file on sd card for better runtime config
59:   * useful so that a clean build is not always necessary
60:   * contains static variables used throughout rest of project
61:   */
62:  class Configuration
63:  {
64:      private:
65:          Configuration();
66:          static Configuration *config_obj;
67:
68:      public:
69:          ~Configuration();
70:
71:          /**
72:           * @return: Configuration -> instance of class to be used throughout program
73:           *
74:           * give user the instance of the singleton class or creates it if it does
75:           * not yet exist
76:           */
77:          static Configuration* get_instance();
78:
79:          pid internal_motor_pid;
80:          pid lift_pid;
81:          pid chassis_pid;
82:
83:          int front_right_port;
84:          int back_left_port;
85:          int front_left_port;
86:          int back_right_port;
87:          int left_intake_port;
88:          int right_intake_port;
89:          int upper_indexer_port;
90:          int lower_indexer_port;
91:
92:          bool front_right_reversed;
93:          bool back_left_reversed;
94:          bool front_left_reversed;
95:          bool back_right_reversed;
96:          bool left_intake_reversed;
97:          bool right_intake_reversed;
98:          bool upper_indexer_reversed;
99:          bool lower_indexer_reversed;
100:
101:         std::vector<int> lift_setpoints;
102:         std::vector<int> tilter_setpoints;
103:         std::vector<int> intake_speeds;
104:
105:         int filter_threshold;
106:         std::string filter_color;  // color to remove
107:
108:
109:         /**
110:          * @return: int -> 1 if file was successfully read, 0 if no changes were made
111:          *
112:          * @see: ../lib/json.hpp
113:          *
```

```
114:         * parses json file looking for data to set variables to
115:         */
116:        int init();
117:
118:
119:        /**
120:         * @return: None
121:         *
122:         * @see: typedef struct pid
123:         *
124:         * prints all the variables in the class
125:         * used for debugging to make sure values are what they are
126:         * supposed to be
127:         */
128:        void print_config_options();
129:
130:    };
131:
132:
133:
134:
135:    #endif
```

```cpp
/**
 * @file: ./RobotCode/src/Configuration.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * @see: Configuration.hpp
 *
 * contains implementation for configuration class
 */

#include <vector>
#include <fstream>

#include "main.h"

#include "../lib/json.hpp"
#include "Configuration.hpp"


Configuration *Configuration::config_obj = NULL;


Configuration::Configuration( )
{
    //set default values for constants in case file can't be read
    internal_motor_pid.kP = 30;
    internal_motor_pid.kI = 37;
    internal_motor_pid.kD = 11;
    internal_motor_pid.I_max = INT32_MAX;

    lift_pid.kP = .1;
    lift_pid.kI = 0.0001;
    lift_pid.kD = 0;
    lift_pid.I_max = INT32_MAX;

    chassis_pid.kP = .0035;
    chassis_pid.kI = 0;
    chassis_pid.kD = 0;
    chassis_pid.I_max = INT32_MAX;

    //536C motor config
    front_right_port = 12;
    back_left_port = 15;
    front_left_port = 16;
    back_right_port = 13;
    left_intake_port = 8;
    right_intake_port = 7;
    upper_indexer_port = 9;
    lower_indexer_port = 17;

    front_right_reversed = 1;
    back_left_reversed = 1;
    front_left_reversed = 0;
    back_right_reversed = 0;
    left_intake_reversed = 0;
    right_intake_reversed = 1;
    upper_indexer_reversed = 0;
    lower_indexer_reversed = 0;

    filter_threshold = 2880;
    filter_color = "blue";


    //536D motor config
    // front_right_port = 13;
    // back_left_port = 1;
    // front_left_port = 20;
    // back_right_port = 19;
    // left_intake_port = 2;
    // right_intake_port = 11;
    // tilter_port = 17;
    // lift_port = 12;
    //
    // front_right_reversed = 1;
    // back_left_reversed = 0;
    // front_left_reversed = 0;
    // back_right_reversed = 1;
    // left_intake_reversed = 0;
    // right_intake_reversed = 1;
    // tilter_reversed = 1;
    // lift_reversed = 0;


    //536A motor config
    // front_right_port = 20;
    // back_left_port = 2;
    // front_left_port = 5;
    // back_right_port = 4;
    // left_intake_port = 8;
    // right_intake_port = 1;
    // tilter_port = 13;
    // lift_port = 11;
    //
    // front_right_reversed = 1;
    // back_left_reversed = 1;
    // front_left_reversed = 1;
    // back_right_reversed = 1;
    // left_intake_reversed = 0;
    // right_intake_reversed = 1;
    // tilter_reversed = 0;
    // lift_reversed = 0;

    std::vector<int> vec1 {100, 300, 400, 500};
    std::vector<int> vec2 {100, 300, 400, 500};
    std::vector<int> vec3 {-63, -30, 0, 30, 63};

    tilter_setpoints = vec1;
    lift_setpoints = vec2;
    intake_speeds = vec3;
}


```

```
114:
115: Configuration::~Configuration( )
116: {
117:
118: }
119:
120:
121:
122: /**
123:  * inits object if object is not already initialized based on a static bool
124:  * sets bool if it is not set
125:  */
126: Configuration* Configuration::get_instance()
127: {
128:     if ( config_obj == NULL )
129:     {
130:         config_obj = new Configuration;
131:     }
132:     return config_obj;
133: }
134:
135:
136: /**
137:  * reads json file into memory in the form of a json object supported by
138:  * a library
139:  * parses json array to get pid constants and setpoints by looking at the size
140:  * sets other variables by looking at their value
141:  */
142: int Configuration::init()
143: {
144:     std::ifstream input("/usd/config.json"); //open file with library
145:     if ( input.fail() )
146:     {
147:         std::cerr << "[ERROR], " << pros::millis() << ", configuration file could not be opened\n";
148:         return 0;
149:     }
150:     nlohmann::json contents;
151:     input >> contents;
152:
153:
154:     std::vector<double> constants1;  //read pid constants for different systems
155:     std::vector<double> constants2;
156:
157:     for ( int i1 = 0; i1 < 4; i1++)
158:     {
159:         double value1 = contents["internal_motor_pid"][i1];
160:         double value2 = contents["lift_pid"][i1];
161:
162:         std::cout << value1 << "\n";
163:         constants1.push_back(value1);
164:         constants2.push_back(value2);
165:     }
166:
167:     internal_motor_pid.kP = constants1.at(0);
168:     internal_motor_pid.kI = constants1.at(1);
169:     internal_motor_pid.kD = constants1.at(2);
170:     internal_motor_pid.I_max = constants1.at(3);
171:
172:     lift_pid.kP = constants2.at(0);
173:     lift_pid.kI = constants2.at(1);
174:     lift_pid.kD = constants2.at(2);
175:     lift_pid.I_max = constants2.at(3);
176:
177:
178:
179:
180:     front_right_port = contents["front_right_port"];  //read motor port definitions
181:     back_left_port = contents["back_left_port"];
182:     front_left_port = contents["front_left_port"];
183:     back_right_port = contents["back_right_port"];
184:     left_intake_port = contents["left_intake_port"];
185:     right_intake_port = contents["right_intake_port"];
186:     upper_indexer_port = contents["upper_indexer_port"];
187:     lower_indexer_port = contents["lower_indexer_port"];
188:
189:     front_right_reversed = contents["front_right_reversed"] == 1 ? true : false; //read motor port reversals
190:     back_left_reversed = contents["back_left_reversed"] == 1 ? true : false;
191:     front_left_reversed = contents["front_left_reversed"] == 1 ? true : false;
192:     back_right_reversed = contents["back_right_reversed"] == 1 ? true : false;
193:     left_intake_reversed = contents["left_intake_reversed"] == 1 ? true : false;
194:     right_intake_reversed = contents["right_intake_reversed"] == 1 ? true : false;
195:     upper_indexer_reversed = contents["upper_indexer_reversed"] == 1 ? true : false;
196:     lower_indexer_reversed = contents["lower_indexer_reversed"] == 1 ? true : false;
197:
198:     filter_threshold = contents["filter_threshold"];
199:
200:     lift_setpoints.clear();
201:     for ( int i2 = 0; i2 < contents["lift_setpoints"].size(); i2++)
202:     {
203:         lift_setpoints.push_back(contents["lift_setpoints"][i2]);
204:
205:     }
206:
207:     intake_speeds.clear();
208:     for ( int i3 = 0; i3 < contents["intake_speeds"].size(); i3++)
209:     {
210:         intake_speeds.push_back(contents["intake_speeds"][i3]);
211:
212:     }
213:
214:     return 1;
215: }
216:
217:
218:
219:
220: /**
221:  * prints all the variables and what they are so that they can be debugged
222:  * makes use of internal pid print function
223:  */
224: void Configuration::print_config_options()
225: {
226:     std::cout << "drive PID constants\n";
```

```
227:     internal_motor_pid.print();
228:     std::cout << "lift PID constants\n";
229:     lift_pid.print();
230: 
231:     std::cout << "\n";
232: 
233:     std::cout << "front_right_port: " << front_right_port << "\n";
234:     std::cout << "back_left_port: " << back_left_port << "\n";
235:     std::cout << "front_left_port: " << front_left_port << "\n";
236:     std::cout << "back_right_port: " << back_right_port << "\n";
237:     std::cout << "left_intake_port: " << left_intake_port << "\n";
238:     std::cout << "right_intake_port: " << right_intake_port << "\n";
239:     std::cout << "upper_indexer_port: " << upper_indexer_port << "\n";
240:     std::cout << "lower_indexer_port: " << lower_indexer_port << "\n";
241: 
242:     std::cout << "front_right_reversed: " << front_right_reversed << "\n";
243:     std::cout << "back_left_reversed: " << back_left_reversed << "\n";
244:     std::cout << "front_left_reversed: " << front_left_reversed << "\n";
245:     std::cout << "back_right_reversed: " << back_right_reversed << "\n";
246:     std::cout << "left_intake_reversed: " << left_intake_reversed << "\n";
247:     std::cout << "right_intake_reversed: " << right_intake_reversed << "\n";
248:     std::cout << "upper_indexer_reversed: " << upper_indexer_reversed << "\n";
249:     std::cout << "lower_indexer_reversed: " << lower_indexer_reversed << "\n";
250: 
251:     std::cout << "\nfilter threshold: " << filter_threshold << "\n";
252: 
253: 
254:     std::cout << "\nlift_setpoints: ";
255:     for ( int i = 0; i < lift_setpoints.size() - 1; i++ )
256:     {
257:         std::cout << lift_setpoints.at(i) << ", ";
258:     }
259:     std::cout << lift_setpoints.at(lift_setpoints.size() - 1) << "\n";
260: 
261: 
262:     std::cout << "\nintake_speeds: ";
263:     for ( int i = 0; i < intake_speeds.size() - 1; i++ )
264:     {
265:         std::cout << intake_speeds.at(i) << ", ";
266:     }
267:     std::cout << intake_speeds.at(intake_speeds.size() - 1) << "\n";
268: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/DriverControl.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * Contains robot move functions. Meant to be run in pros task
8:   *
9:   */
10:
11: #ifndef __DRIVERCONTROL_HPP__
12: #define __DRIVERCONTROL_HPP__
13:
14: #include <cstdlib>
15:
16: #include "../include/main.h"
17:
18: #include "objects/subsystems/chassis.hpp"
19: #include "objects/controller/controller.hpp"
20: #include "objects/motors/Motors.hpp"
21:
22: #define AUTON_DEBUG
23:
24:
25: /**
26:  * @param: void* -> not used
27:  * @return: None
28:  *
29:  * @see: Motors.hpp
30:  * @see: Controller.hpp
31:  *
32:  * meant to be run on task
33:  * function cycles through and allows user to controll robot
34:  *
35:  */
36: void driver_control(void*);
37:
38:
39: #endif
```

```
 1:
 2: /**
 3:  * @file: ./RobotCode/src/DriverControl.cpp
 4:  * @author: Aiden Carney
 5:  * @reviewed_on: 10/15/2019
 6:  * @reviewed_by: Aiden Carney
 7:  *
 8:  * @see: DriverControl.hpp
 9:  *
10:  */
11:
12: #include <cstdlib>
13: #include <cmath>
14:
15: #include "../include/main.h"
16:
17: #include "objects/lcdCode/gui.hpp"
18: #include "objects/subsystems/chassis.hpp"
19: #include "objects/subsystems/Indexer.hpp"
20: #include "objects/subsystems/intakes.hpp"
21: #include "objects/controller/controller.hpp"
22: #include "objects/motors/Motors.hpp"
23: #include "objects/sensors/Sensors.hpp"
24: #include "Configuration.hpp"
25: #include "DriverControl.hpp"
26: #include "Configuration.hpp"
27:
28:
29:
30: /**
31:  * uses if statements to control motor based on controller settings
32:  * checks to set it to zero based on if static var in Motors class allows it
33:  * this is to make sure that other tasks can controll Motors too
34:  */
35: void driver_control(void*)
36: {
37:     Configuration *config = Configuration::get_instance();
38:
39:     Controller controllers;
40:
41:     Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
42:     Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, config->filter_color);
43:     Intakes intakes(Motors::left_intake, Motors::right_intake);
44:
45:     int left_analog_y = 0;
46:     int right_analog_y = 0;
47:
48:     bool auto_filter = true;
49:     bool hold_intakes_out = true;
50:     int intake_start_time = 0;  // no possible way to think indexer should run at the start of driver control
51:
52:     controllers.master.print(0, 0, "Auto Filter %s     ", config->filter_color);
53:
54:     while ( true ) {
55:         controllers.update_button_history();
56:
57:     // section for front roller intake movement
58:         if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_R1)) {  // define velocity for main intake
59:             intakes.intake();
60:             intake_start_time = pros::millis();
61:         } else if(hold_intakes_out){  // rest state is outward with motor power
62:             intakes.hold_outward();
63:         } else {  // rest state is no motor power
64:             intakes.stop();
65:         }
66:
67:         if(controllers.btn_get_release(pros::E_CONTROLLER_DIGITAL_R2)) {
68:             hold_intakes_out = !hold_intakes_out;
69:         }
70:
71:     // section for indexer motion
72:         if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_L1) && auto_filter) {  // define movement for indexer subsystem
73:             indexer.auto_index();
74:         } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_L1) && !auto_filter) {
75:             indexer.index();
76:         } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_LEFT)) {
77:             indexer.filter();
78:         } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_L2) && auto_filter) {
79:             indexer.auto_increment();
80:         } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_L2) && !auto_filter) {
81:             indexer.increment();
82:         } else if(controllers.master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_RIGHT)) {
83:             indexer.fix_ball(true);
84:         } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_X)) {
85:             indexer.index_no_backboard();
86:         } else if (pros::millis() < intake_start_time + 1000) {
87:             indexer.auto_increment();
88:         } else {
89:             indexer.hard_stop();
90:         }
91:
92:         if(controllers.btn_get_release(pros::E_CONTROLLER_DIGITAL_LEFT)) {
93:             auto_filter = !auto_filter;
94:             if(auto_filter) {  // give different message if not auto filtering
95:                 controllers.master.print(0, 0, "Auto Filter %s     ", config->filter_color);
96:             } else {
97:                 controllers.master.print(0, 0, "Man Filter %s     ", config->filter_color);
98:             }
99:         }
100:
101:     // section for setting filter color
102:         if(controllers.btn_get_release(pros::E_CONTROLLER_DIGITAL_A)) {  // cycle filter colors
103:             if(config->filter_color == "red") {
104:                 config->filter_color = "blue";
105:             } else if(config->filter_color == "blue") {
106:                 config->filter_color = "none";
107:             } else if(config->filter_color == "none") {
108:                 config->filter_color = "red";
109:             }
110:
111:             if(auto_filter) {  // give different message if not auto filtering
112:                 controllers.master.print(0, 0, "Auto Filter %s     ", config->filter_color);
113:             } else {
```

```
114:                 controllers.master.print(0, 0, "Man Filter %s     ", config->filter_color);
115:             }
116:             std::cout << "filtering " << config->filter_color << "\n";
117:             indexer.update_filter_color(config->filter_color);
118:         }
119:
120:
121:     // section for chassis movement
122:         if(std::abs(controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y)) < 5) {   // define deadzone for left analog input on the y axis
123:             left_analog_y = 0;
124:         } else {
125:             left_analog_y = controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y);
126:         }
127:
128:         if(std::abs(controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y)) < 5) {   // define deadzone for right analog input on the y axis
129:             right_analog_y = 0;
130:         } else {
131:             right_analog_y = controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y);
132:         }
133:
134:         // float corrected_speed = ( .0000433264318666017 * std::pow( leftDriveSpeed, 3 ) ) + ( 0.29594689028631 * leftDriveSpeed);
135:         Motors::front_left.user_move(left_analog_y);
136:         Motors::back_left.user_move(left_analog_y);
137:
138:         // float corrected_speed = ( .0000433264318666017 * std::pow( rightDriveSpeed, 3 ) ) + ( 0.29594689028631 * rightDriveSpeed);
139:         Motors::front_right.user_move(right_analog_y);
140:         Motors::back_right.user_move(right_analog_y);
141:
142:
143:         if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_DOWN) ) {
144:             Autons auton;
145:             auton.deploy();
146:         }
147:
148:         pros::delay(5);
149:
150:     }
151: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/controller/controller.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 11/8/19
5:   * @reviewed_by: Aiden Carney
6:   * TODO: remove static members and replace with singleton class
7:   *
8:   * contains class that has data about controllers
9:   */
10:
11:  #include <unordered_map>
12:  #include <string>
13:  #include <deque>
14:
15:  #include "../../../include/main.h"
16:  #include "../../../include/api.h"
17:  #include "../../../include/pros/rtos.hpp"
18:  #include "../../../include/pros/motors.hpp"
19:
20:  #ifndef __CONTROLLER_HPP__
21:  #define __CONTROLLER_HPP__
22:
23:  /**
24:   * contains controller objects as well as unordered maps that hold information
25:   * about what the controller does
26:   * TODO: move unordered maps to configuration and make more robust so it does not have to be updated
27:   */
28:  class Controller
29:  {
30:      private:
31:          static std::unordered_map <pros::controller_digital_e_t, std::deque<bool>* > master_btn_history;
32:          static std::unordered_map <pros::controller_digital_e_t, std::deque<bool>* > partner_btn_history;
33:
34:      public:
35:          Controller();
36:          ~Controller();
37:
38:          static pros::Controller master;
39:          static pros::Controller partner;
40:
41:          static std::unordered_map <pros::controller_analog_e_t, std::string> MASTER_CONTROLLER_ANALOG_MAPPINGS;
42:          static std::unordered_map <pros::controller_analog_e_t, std::string> PARTNER_CONTROLLER_ANALOG_MAPPINGS;
43:          static std::unordered_map <pros::controller_digital_e_t, std::string> MASTER_CONTROLLER_DIGITAL_MAPPINGS;
44:          static std::unordered_map <pros::controller_digital_e_t, std::string>  PARTNER_CONTROLLER_DIGITAL_MAPPINGS;
45:
46:          void update_button_history();
47:
48:          bool btn_get_release(pros::controller_digital_e_t btn, int controller=0);
49:          bool btn_get_start_press(pros::controller_digital_e_t btn, int controller=0);
50:          bool btn_is_pressing(pros::controller_digital_e_t btn, int controller=0);
51:
52:  };
53:
54:
55:
56:
57:  #endif
```

```
/**
 * @file: ./RobotCode/src/controller/controller.cpp
 * @author: Aiden Carney
 * @reviewed_on: 11/8/19
 * @reviewed_by: Aiden Carney
 *
 * @see: controller.hpp
 *
 * contains definitions for static members of class
 */

#include <unordered_map>
#include <string>

#include "../../../include/main.h"
#include "../../../include/api.h"
#include "../../../include/pros/rtos.hpp"
#include "../../../include/pros/motors.hpp"

#include "controller.hpp"



std::unordered_map <pros::controller_digital_e_t, std::deque<bool>* > Controller::master_btn_history = {
    {pros::E_CONTROLLER_DIGITAL_L1, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_L2, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_R2, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_R1, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_UP, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_DOWN, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_LEFT, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_RIGHT, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_X, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_B, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_Y, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_A, new std::deque<bool>}
};

std::unordered_map <pros::controller_digital_e_t, std::deque<bool>* > Controller::partner_btn_history = {
    {pros::E_CONTROLLER_DIGITAL_L1, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_L2, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_R2, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_R1, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_UP, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_DOWN, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_LEFT, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_RIGHT, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_X, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_B, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_Y, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_A, new std::deque<bool>}
};

pros::Controller Controller::master(pros::E_CONTROLLER_MASTER);
pros::Controller Controller::partner(pros::E_CONTROLLER_PARTNER);

//mappings for each controller
std::unordered_map <pros::controller_analog_e_t, std::string> Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS = {
    {pros::E_CONTROLLER_ANALOG_LEFT_X, "None"},
    {pros::E_CONTROLLER_ANALOG_LEFT_Y, "Left Side Chassis"},
    {pros::E_CONTROLLER_ANALOG_RIGHT_X, "None"},
    {pros::E_CONTROLLER_ANALOG_RIGHT_Y, "Right Side Chassis"}
};

std::unordered_map <pros::controller_analog_e_t, std::string> Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS = {
    {pros::E_CONTROLLER_ANALOG_LEFT_X, "None"},
    {pros::E_CONTROLLER_ANALOG_LEFT_Y, "None"},
    {pros::E_CONTROLLER_ANALOG_RIGHT_X, "None"},
    {pros::E_CONTROLLER_ANALOG_RIGHT_Y, "None"}
};

std::unordered_map <pros::controller_digital_e_t, std::string> Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS = {
    {pros::E_CONTROLLER_DIGITAL_L1, "lift up"},
    {pros::E_CONTROLLER_DIGITAL_L2, "lift down"},
    {pros::E_CONTROLLER_DIGITAL_R2, "Outake"},
    {pros::E_CONTROLLER_DIGITAL_R1, "Intake"},
    {pros::E_CONTROLLER_DIGITAL_UP, "Tilter up"},
    {pros::E_CONTROLLER_DIGITAL_DOWN, "Tilter down"},
    {pros::E_CONTROLLER_DIGITAL_LEFT, "None"},
    {pros::E_CONTROLLER_DIGITAL_RIGHT, "None"},
    {pros::E_CONTROLLER_DIGITAL_X, "auto deploy"},
    {pros::E_CONTROLLER_DIGITAL_B, "Toggle brakes"},
    {pros::E_CONTROLLER_DIGITAL_Y, "auto dump"},
    {pros::E_CONTROLLER_DIGITAL_A,  "run auton"}
};

std::unordered_map <pros::controller_digital_e_t, std::string>  Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS = {
    {pros::E_CONTROLLER_DIGITAL_L1, "None"},
    {pros::E_CONTROLLER_DIGITAL_L2, "None"},
    {pros::E_CONTROLLER_DIGITAL_R2, "None"},
    {pros::E_CONTROLLER_DIGITAL_R1, "None"},
    {pros::E_CONTROLLER_DIGITAL_UP, "None"},
    {pros::E_CONTROLLER_DIGITAL_DOWN, "None"},
    {pros::E_CONTROLLER_DIGITAL_LEFT, "None"},
    {pros::E_CONTROLLER_DIGITAL_RIGHT, "None"},
    {pros::E_CONTROLLER_DIGITAL_X, "None"},
    {pros::E_CONTROLLER_DIGITAL_B, "None"},
    {pros::E_CONTROLLER_DIGITAL_Y, "None"},
    {pros::E_CONTROLLER_DIGITAL_A,  "None"}
};




Controller::Controller()
{

}

Controller::~Controller()
{

}
```

```
114:
115:
116: void Controller::update_button_history()
117: {
118:     for (std::pair<pros::controller_digital_e_t, std::deque<bool>*> element : master_btn_history) {
119:         element.second->push_back(master.get_digital(element.first));
120:         if(element.second->size() > 3) {
121:             element.second->pop_front();
122:         }
123:
124:         while(element.second->size() < 3) {  // forces deque size to be no less than 3
125:             element.second->push_back(master.get_digital(element.first));
126:         }
127:     }
128:
129:     for (std::pair<pros::controller_digital_e_t, std::deque<bool>*> element : partner_btn_history) {
130:         element.second->push_back(master.get_digital(element.first));
131:         if(element.second->size() > 3) {
132:             element.second->pop_front();
133:         }
134:
135:         while(element.second->size() < 3) {  // forces deque size to be no less than 3
136:             element.second->push_back(master.get_digital(element.first));
137:         }
138:     }
139: }
140:
141:
142: bool Controller::btn_get_release(pros::controller_digital_e_t btn, int controller /** 0 **/) {
143:     bool pressed_and_released = false;
144:     if(!controller) {
145:         if(master_btn_history.at(btn)->at(1) and !master_btn_history.at(btn)->at(2)) {
146:             pressed_and_released = true;
147:         }
148:     }
149:     else {
150:         if(partner_btn_history.at(btn)->at(1) and !partner_btn_history.at(btn)->at(2)) {
151:             pressed_and_released = true;
152:         }
153:     }
154:     return pressed_and_released;
155: }
156:
157:
158: bool Controller::btn_get_start_press(pros::controller_digital_e_t btn, int controller /** 0 **/) {
159:     bool press_start = false;
160:     if(!controller) {
161:         if(!master_btn_history.at(btn)->at(1) and master_btn_history.at(btn)->at(2)) {
162:             press_start = true;
163:         }
164:     }
165:     else {
166:         if(!partner_btn_history.at(btn)->at(1) and partner_btn_history.at(btn)->at(2)) {
167:             press_start = true;
168:         }
169:     }
170:
171:     return press_start;
172: }
173:
174:
175: bool Controller::btn_is_pressing(pros::controller_digital_e_t btn, int controller /** 0 **/) {
176:     bool pressing = false;
177:     if(!controller) {
178:         pressing = master_btn_history.at(btn)->at(2);
179:     }
180:     else {
181:         pressing = partner_btn_history.at(btn)->at(2);
182:     }
183:
184:     return pressing;
185: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/motors/Motor.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 2/16/2020
5:   * @reviewed_by: Aiden Carney
6:   * TODO:
7:   *
8:   * contains a wrapper class for a pros::Motor
9:   */
10:
11: #ifndef __MOTOR_HPP__
12: #define __MOTOR_HPP__
13:
14: #include <atomic>
15:
16: #include "main.h"
17:
18: #include "../../Configuration.hpp"
19:
20:
21: typedef enum {
22:     e_builtin_velocity_pid,
23:     e_voltage,
24:     e_custom_velocity_pid
25: } motor_mode;
26:
27: /**
28:  * @see: pros::Motor
29:  * @see: ../../Configuration.hpp
30:  *
31:  * wrapper class for pros::Motor
32:  * contains implementation for better runtime port configuration
33:  * contains easier implementation for slew rate control
34:  * contains a pid velocity controller that can be enabled for consistent motor output
35:  */
36: class Motor
37: {
38:     private:
39:         int motor_port;
40:
41:         pros::Motor *motor;
42:
43:         int log_level;
44:
45:         bool slew_enabled;
46:         int slew_rate;
47:
48:         bool velocity_pid_enabled;
49:         int prev_velocity;
50:         pid internal_motor_pid;
51:         double integral;
52:         double prev_error;
53:
54:         motor_mode mode = e_voltage;
55:         int voltage_setpoint;
56:         int prev_voltage_setpoint;
57:         int velocity_setpoint;
58:
59:
60:         int to_voltage(int velocity);
61:         int to_velocity(int voltage);
62:
63:         /**
64:          * @param: int voltage -> the voltage to set the motor to
65:          * @return: int -> if setting motor voltage was successful or not
66:          *
67:          * @see: pros::Motor
68:          *
69:          * sets voltage of motor on interval [-12000,12000]
70:          */
71:         int set_voltage_setpoint( int voltage );
72:
73:
74:         int set_velocity_setpoint(int new_velocity);
75:
76:
77:         /**
78:          * @param int target -> the new voltage that could be requested
79:          * @param int previous -> the previous voltage to calculate change in voltage over time
80:          * @param in delta_t -> the time that has elapsed
81:          * @return: int -> the rate of the voltage set based on time elapsed and previous voltage
82:          *
83:          * @see: set_voltage_setpoint()
84:          *
85:          * calculates the rate of change of the voltage (mv/ms) that the new
86:          * voltage is trying to reach
87:          */
88:         int calc_target_rate( int target, int previous, int delta_t );
89:
90:         /**
91:          * @param: int voltage -> a possible motor voltage in mv on interval [-12000,12000]
92:          * @return: int -> the corresponding velocity for a given voltage
93:          *
94:          * TODO: add checking if the voltage is not on the interval
95:          *
96:          * calculates the corresponding velocity for a given voltage in mv
97:          * the velocity range corresponds to the gearset of the motor
98:          * velocity ranges are ~20% higher than what they are rated for
99:          * because motors can achieve this velocity when supplied 12V
100:         */
101:        int calc_target_velocity( int voltage );
102:
103:        /**
104:         * @return: int -> the voltage that the motor will be set at
105:         *
106:         * @see: slew rate functions contained in this class
107:         *
108:         * returns the target voltage based on the voltage set by the user
109:         * but that is either increased or decreased by the velocity pid if that
110:         * is enabled, or the slew rate code which limits the rate that the
111:         * voltage can increase
112:         */
113:        int get_target_voltage( int delta_t );
```

```
114:
115:
116:        std::atomic<bool> lock;  //protect motor functions from concurrent access
117:        bool allow_driver_control;
118:
119:
120:    public:
121:        Motor(int port, pros::motor_gearset_e_t gearset, bool reversed);
122:        Motor(int port, pros::motor_gearset_e_t gearset, bool reversed, pid pid_consts);
123:        ~Motor();
124:
125:    //accessor functions
126:
127:        /**
128:         * @return: double -> the actual velocity of the motor
129:         *
130:         * @see: pros::Motor
131:         *
132:         * returns the actual velocity of the motor as calculated internally by
133:         * the pros::Motor
134:         */
135:        double get_actual_velocity( );
136:
137:        /**
138:         * @return: double -> the actual voltage of the motor
139:         *
140:         * @see: pros::Motor
141:         *
142:         * returns the actual voltage of the motor as calculated internally by
143:         * the pros::Motor
144:         */
145:        double get_actual_voltage( );
146:
147:        /**
148:         * @return: int -> the actual current being supplied to the motor
149:         *
150:         * @see: pros::Motor
151:         *
152:         * returns the actual current being supplied to the motor as calculated internally by
153:         * the pros::Motor
154:         */
155:        int get_current_draw( );
156:
157:        /**
158:         * @return: double -> the encoder value of the motor
159:         *
160:         * @see: pros::Motor
161:         *
162:         * returns the encoder position of the motor in degrees as calculated internally by
163:         * the pros::Motor
164:         */
165:        double get_encoder_position( );
166:
167:        /**
168:         * @return: pros::motor_gearset_e_t -> the gearing of the motor
169:         *
170:         * @see: pros::Motor
171:         *
172:         * returns the gearset internally used by the motor per the pros::Motor
173:         */
174:        pros::motor_gearset_e_t get_gearset( );
175:
176:        /**
177:         * @return: pros::motor_brake_mode_e_t -> the brakemode of the motor
178:         *
179:         * @see: pros::Motor
180:         *
181:         * returns the brakemode internally used by the motor per the pros::Motor
182:         */
183:        pros::motor_brake_mode_e_t get_brake_mode( );
184:
185:        /**
186:         * @return: int -> the port of the motor
187:         *
188:         * returns the port that the motor is set on
189:         */
190:        int get_port( );
191:
192:        /**
193:         * @return: pid -> struct of pid constants
194:         *
195:         * returns the pid constants in use by the motor
196:         */
197:        pid get_pid( );
198:
199:        /**
200:         * @return: int -> the slew rate in use by the motor
201:         *
202:         * returns the slew rate in mV/ms in use by the motor
203:         */
204:        int get_slew_rate( );
205:
206:        /**
207:         * @return: double -> the power drawn by the motor
208:         *
209:         * @see: pros::Motor
210:         *
211:         * returns the power that the motor is drawing in Watts
212:         */
213:        double get_power( );
214:
215:        /**
216:         * @return: double -> the temperature of the motor
217:         *
218:         * @see: pros::Motor
219:         *
220:         * returns the temperature of the motor in degrees C
221:         */
222:        double get_temperature( );
223:
224:        /**
225:         * @return: double -> the torque output of the motor
226:         *
```

```
227:         * @see: pros::Motor
228:         *
229:         * returns the torque output of the motor in Nm
230:         */
231:        double get_torque( );
232:
233:        /**
234:         * @return: int -> the direction the motor is spinning
235:         *
236:         * @see: pros::Motor
237:         *
238:         * returns the direction of the motor
239:         * 1 for moving in the positive direction
240:         * -1 for moving in the negative direction
241:         */
242:        int get_direction( );
243:
244:        /**
245:         * @return: int -> the efficiency of the motor
246:         *
247:         * @see: pros::Motor
248:         *
249:         * returns the efficiency of the motor as a percentage
250:         */
251:        int get_efficiency( );
252:
253:        /**
254:         * @return: int -> if the motor is a rest
255:         *
256:         * @see: pros::Motor
257:         *
258:         * returns 1 if the motor is not moving and 0 if the motor is moving
259:         */
260:        int is_stopped( );
261:
262:        /**
263:         * @return: int -> if the motor has been reversed or not
264:         *
265:         * @see: pros::Motor
266:         *
267:         * returns 1 if the motor has been reversed and 0 if the motor was not reversed
268:         */
269:        int is_reversed( );
270:
271:
272:
273:
274:    //setter functions
275:        /**
276:         * @param: int port -> the new port for the motor
277:         * @return: int -> if the change was successful or not
278:         *
279:         * @see: pros::Motor
280:         *
281:         * returns 1 on success
282:         */
283:        int set_port( int port );
284:
285:        /**
286:         * @return: int -> if function was successful or not
287:         *
288:         * @see: pros::Motor
289:         *
290:         * returns 1 on success
291:         */
292:        int tare_encoder( );
293:
294:        /**
295:         * @param: pros::motor_brake_mode_e_t -> the new brake mode for the motor
296:         * @return: int -> if the change was successful or not
297:         *
298:         * @see: pros::Motor
299:         *
300:         * returns 1 on success
301:         */
302:        int set_brake_mode( pros::motor_brake_mode_e_t brake_mode );
303:
304:        /**
305:         * @param: pros::motor_gearset_e_t -> the new gearset for the motor
306:         * @return: int -> if the change was successful or not
307:         *
308:         * @see: pros::Motor
309:         *
310:         * returns 1 on success
311:         */
312:        int set_gearing( pros::motor_gearset_e_t gearset );
313:
314:        /**
315:         * @return: int -> if motor was reversed or not
316:         *
317:         * @see: pros::Motor
318:         *
319:         * returns 1 on success
320:         */
321:        int reverse_motor( );
322:
323:        /**
324:         * @param: pid pid_consts -> the new pid constants for the motor
325:         * @return: int -> if the change was successful or not
326:         *
327:         * @see: pros::Motor
328:         *
329:         * returns 1 on success
330:         */
331:        int set_pid( pid pid_consts );
332:
333:        /**
334:         * @param: int logging -> the new log level, 0-5, 5 is most verbose
335:         * @return: None
336:         *
337:         * @see: pros::Motor
338:         *
339:         * updates how verbose the logging is, 0 is no logging, 5 is very
```

```
340:         * verbose
341:         */
342:        void set_log_level( int logging );
343:
344:
345:
346:
347:    //movement functions
348:        /**
349:         * @param: int voltage -> the voltage to set the motor to
350:         * @return: int -> if setting motor voltage was successful or not
351:         *
352:         * @see: pros::Motor
353:         *
354:         * takes range [-127,127] and scales it to [-12000,12000]
355:         * makes it easier to map to controller input
356:         */
357:        int move( int voltage );
358:
359:        /**
360:         * @param: int voltage -> the voltage to set the motor to
361:         * @return: int -> if the motor voltage was set or not
362:         *
363:         * @see: pros::Motor
364:         *
365:         * takes range [-127,127] and scales it to [-12000,12000]
366:         * makes it easier to map to controller input. This function acts as
367:         * a wrapper to move but with a check to see if the driver control lock
368:         * is taken. Use this function when using the controller in driver control
369:         * to set the motor value.
370:         */
371:        int user_move( int voltage );
372:
373:        /**
374:         * @param: int velocity -> the velocity to set the motor to
375:         * @return: int -> if setting motor velocity was successful or not
376:         *
377:         * @see: pros::Motor
378:         *
379:         * takes range [-gearset_min + ~20%, gearset_min + ~20%] and scales it
380:         * to [-12000,12000]
381:         * used to make motor performance more consistent when velocity pid is
382:         * enabled
383:         * doesn't use built in pid because max motor ouput is limited by
384:         * approximately 20%
385:         */
386:        int move_velocity( int velocity );
387:
388:        int set_voltage(int voltage);
389:
390:
391:    //slew rate control functions
392:        /**
393:         * @param: int rate -> the new slew rate in mv/ms
394:         * @return: int -> 1 on success
395:         *
396:         * @see: pros::Motor
397:         *
398:         * sets the new rate that the voltage can increase at
399:         * used for either acceleration control for less wheel slippage
400:         * or to protect motors from voltage spikes
401:         */
402:        int set_slew( int rate );
403:
404:        /**
405:         * @return: None
406:         *
407:         * @see: pros::Motor
408:         *
409:         * sets slew rate code to be used to limit voltage change rate
410:         */
411:        void enable_slew( );
412:
413:        /**
414:         * @return: None
415:         *
416:         * @see: pros::Motor
417:         *
418:         * sets slew rate code to not be used to limit voltage change rate
419:         */
420:        void disable_slew( );
421:
422:
423:
424:
425:    //velocity pid control functions
426:        /**
427:         * @return: None
428:         *
429:         * @see: pros::Motor
430:         *
431:         * sets new mode for the motor to follow
432:         */
433:        void set_motor_mode(motor_mode new_mode);
434:
435:
436:
437:    //driver control lock setting and clearing functions
438:        /**
439:         * @return: None
440:         *
441:         * @see: pros::Motor
442:         *
443:         * sets a lock that can be used to prevent controller from being able
444:         * to set motor voltage
445:         */
446:        void enable_driver_control( );
447:
448:        /**
449:         * @return: None
450:         *
451:         * @see: pros::Motor
452:         *
```

```
453:         * clears a lock that can be used to prevent controller from being able
454:         * to set motor voltage
455:         */
456:        void disable_driver_control( );
457:
458:        /**
459:         * @return: int -> if driver control lock is cleared
460:         *
461:         * @see: pros::Motor
462:         *
463:         * returns 1 if lock is cleared, 0 otherwise
464:         */
465:        int driver_control_allowed( );
466:
467:
468:      //function to run on thread
469:        /**
470:         * @param: int delta_t -> the amount of time elapsed since the last time the function was called
471:         * @return: int -> the voltage to set the motor to
472:         *
473:         * @see: pros::Motor
474:         *
475:         * used to be run on long living thread so that motor can have PID and
476:         * slew rate code built into it easier
477:         * also contains logging implementation and adds to logger queue
478:         */
479:        int run( int delta_t );
480:  };
481:
482:
483:
484:
485:  #endif
```

```
/**
 * @file: ./RobotCode/src/objects/motors/Motor.cpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 * TODO: Clean up how logging message is set
 *
 * contains a implementation for wrapper class for a pros::Motor
 */

#include <atomic>

#include "main.h"

#include "../../Configuration.hpp"
#include "../serial/Logger.hpp"
#include "Motor.hpp"



Motor::Motor( int port, pros::motor_gearset_e_t gearset, bool reversed )
{
    lock = ATOMIC_VAR_INIT(false);
    allow_driver_control = true;

    while ( lock.exchange( true ) ); //aquire motor lock

    motor_port = port;

    motor = new pros::Motor(port, gearset, reversed, pros::E_MOTOR_ENCODER_DEGREES);

    prev_velocity = 0;

    log_level = 0;

    slew_enabled = false;  // default slew rate to false
    slew_rate = 30;  //approx. 5% voltage per 20ms == 400ms to reach full voltage

    prev_voltage_setpoint = 0;
    voltage_setpoint = 0;
    velocity_setpoint = 0;

    Configuration *configuration = Configuration::get_instance();
    internal_motor_pid.kP = configuration->internal_motor_pid.kP;
    internal_motor_pid.kI = configuration->internal_motor_pid.kI;
    internal_motor_pid.kD = configuration->internal_motor_pid.kD;
    internal_motor_pid.I_max = configuration->internal_motor_pid.I_max;
    integral = 0;
    prev_error = 0;

    lock.exchange(false);
}


Motor::Motor(int port, pros::motor_gearset_e_t gearset, bool reversed, pid pid_consts)
{
    lock = ATOMIC_VAR_INIT(false);
    allow_driver_control = true;

    while ( lock.exchange( true ) ); //aquire motor lock

    motor_port = port;

    motor = new pros::Motor(port, gearset, reversed, pros::E_MOTOR_ENCODER_DEGREES);

    prev_velocity = 0;

    log_level = 0;

    slew_enabled = false;
    slew_rate = 30;  //approx. 5% voltage per 20ms == 400ms to reach full voltage

    prev_voltage_setpoint = 0;
    voltage_setpoint = 0;
    velocity_setpoint = 0;

    internal_motor_pid.kP = pid_consts.kP;
    internal_motor_pid.kI = pid_consts.kI;
    internal_motor_pid.kD = pid_consts.kD;
    internal_motor_pid.I_max = pid_consts.I_max;
    integral = 0;
    prev_error = 0;

    lock.exchange(false);
}


Motor::~Motor( )
{
    delete motor;
}



int Motor::to_voltage(int velocity) {
    pros::motor_gearset_e_t gearset = motor->get_gearing();

    int prev_max;
    int prev_min;

    if ( gearset == pros::E_MOTOR_GEARSET_36 )  //100 RPM Motor
    {
        prev_max = 120;
        prev_min = -120;
    }
    if ( gearset == pros::E_MOTOR_GEARSET_06 ) //600 RPM Motor
    {
        prev_max = 720;
        prev_min = -720;
    }
    else //default to 200 RPM motor because that is most commonly used
    {
        prev_max = 240;
```

```
114:         prev_min = -240;
115:     }
116:
117:     int new_max = 12000;
118:     int new_min = -12000;
119:
120:     int voltage = (((velocity - prev_min) * (new_max - new_min)) / (prev_max - prev_min)) + new_min;
121:
122:     return voltage;
123: }
124:
125:
126: int Motor::to_velocity(int voltage) {
127:     int prev_max = 12000;
128:     int prev_min = -12000;
129:
130:     pros::motor_gearset_e_t gearset = motor->get_gearing();
131:
132:     int new_max;
133:     int new_min;
134:
135:     if ( gearset == pros::E_MOTOR_GEARSET_36 )  //100 RPM Motor
136:     {
137:         new_max = 120;
138:         new_min = -120;
139:     }
140:     if ( gearset == pros::E_MOTOR_GEARSET_06 ) //600 RPM Motor
141:     {
142:         new_max = 720;
143:         new_min = -720;
144:     }
145:     else //default to 200 RPM motor because that is most commonly used
146:     {
147:         new_max = 240;
148:         new_min = -240;
149:     }
150:
151:     int velocity = (((voltage - prev_min) * (new_max - new_min)) / (prev_max - prev_min)) + new_min;
152:
153:
154:     return velocity;
155: }
156:
157:
158: /**
159:  * calculates the rate that the motor would be set to with a target the previous
160:  * voltage, and how much time has passed
161:  * returns rate in mv/ms
162:  */
163: int Motor::calc_target_rate( int target, int previous, int delta_t)
164: {
165:     int delta_v = target - previous;
166:     int rate;
167:     if ( delta_t == 0 && delta_v == 0 )
168:     {
169:         rate = 0;
170:     }
171:     else if ( delta_t == 0 && delta_v != 0 )
172:     {
173:         rate = INT32_MAX;  //essentially undefined but still represented as integer
174:     }
175:     else
176:     {
177:         rate = delta_v / delta_t;
178:     }
179:
180:     return rate;
181: }
182:
183:
184: /**
185:  * returns the target voltage set to the motor after performing PID and slew rate
186:  * calculations on it
187:  */
188: int Motor::get_target_voltage( int delta_t )
189: {
190:     double kP = internal_motor_pid.kP;
191:     double kI = internal_motor_pid.kI;
192:     double kD = internal_motor_pid.kD;
193:     double I_max = internal_motor_pid.I_max;
194:
195:     int voltage;
196:     int calculated_target_voltage = voltage_setpoint;
197:
198:     //velocity pid is enabled when the target voltage does not change
199:     if ( mode == e_custom_velocity_pid && voltage_setpoint == prev_voltage_setpoint )
200:     {
201:         int error =  to_velocity(voltage_setpoint) - get_actual_velocity();
202:         if ( std::abs(integral) > I_max )
203:         {
204:             integral = 0;
205:         }
206:         else
207:         {
208:             integral = integral + error;
209:         }
210:         double derivative = error - prev_error;
211:         prev_error = error;
212:
213:
214:         calculated_target_voltage = (kP * error) + (kI * integral) + (kD * derivative);
215:     }
216:
217:     //ensure that voltage range is allowed by the slew rate set
218:     int rate = calc_target_rate(calculated_target_voltage, get_actual_voltage(), delta_t);
219:     if ( slew_enabled && std::abs(rate) > slew_rate )
220:     {
221:         int max_delta_v = slew_rate * delta_t;
222:         int polarity = 1;               // rate will be positive or negative if motor is gaining
223:         if ( rate < 0 )                 // or losing velocity
224:         {                           // the polarity ensures that the max voltage is added
225:             polarity = -1;              // in the correct direction so that the motor's velocity
226:         }                           // will increase in the correct direction
```

```
227:
228:        voltage = get_actual_voltage() + (polarity * max_delta_v);
229:    }
230:    else if ( voltage_setpoint == 0 )
231:    {
232:        voltage = 0;
233:    }
234:    else if ( calculated_target_voltage != 0 )
235:    {
236:        voltage = calculated_target_voltage;
237:    }
238:
239:    prev_voltage_setpoint = voltage_setpoint;
240:
241:
242:    return voltage;
243: }
244:
245:
246:
247:
248: //accessor functions
249:
250: /**
251:  * returns velocity of motor
252:  */
253: double Motor::get_actual_velocity( )
254: {
255:    return motor->get_actual_velocity();
256: }
257:
258:
259: /**
260:  * returns voltage of motor
261:  */
262: double Motor::get_actual_voltage( )
263: {
264:    return motor->get_voltage();
265: }
266:
267:
268: /**
269:  * returns current drawn by motor in mA
270:  */
271: int Motor::get_current_draw( )
272: {
273:    return motor->get_current_draw();
274: }
275:
276:
277: /**
278:  * returns encoder position of motor in degrees
279:  */
280: double Motor::get_encoder_position( )
281: {
282:    return motor->get_position();
283: }
284:
285:
286: /**
287:  * returns gearset of motor
288:  */
289: pros::motor_gearset_e_t Motor::get_gearset( )
290: {
291:    return motor->get_gearing();
292: }
293:
294:
295: /**
296:  * returns brake mode of motor
297:  */
298: pros::motor_brake_mode_e_t Motor::get_brake_mode( )
299: {
300:    return motor->get_brake_mode();
301: }
302:
303:
304: /**
305:  * returns port of motor
306:  */
307: int Motor::get_port( )
308: {
309:    return motor_port;
310: }
311:
312:
313: /**
314:  * returns pid constants used by motor
315:  */
316: pid Motor::get_pid( )
317: {
318:    return internal_motor_pid;
319: }
320:
321:
322: /**
323:  * returns slew rate used by motor
324:  */
325: int Motor::get_slew_rate( )
326: {
327:    return slew_rate;
328: }
329:
330:
331: /**
332:  * returns power of motor in watts
333:  */
334: double Motor::get_power( )
335: {
336:    return motor->get_power();
337: }
338:
339:
```

```
340:  /**
341:   * returns temperature of motor in degrees C
342:   */
343:  double Motor::get_temperature( )
344:  {
345:     return motor->get_temperature();
346:  }
347:
348:
349:  /**
350:   * returns torque of motor in Nm
351:   */
352:  double Motor::get_torque( )
353:  {
354:     return motor->get_torque();
355:  }
356:
357:
358:  /**
359:   * returns direction motor is spinning
360:   */
361:  int Motor::get_direction( )
362:  {
363:     return motor->get_direction();
364:  }
365:
366:
367:  /**
368:   * returns efficiency of motor as a percent
369:   */
370:  int Motor::get_efficiency( )
371:  {
372:     return motor->get_efficiency();
373:  }
374:
375:
376:  /**
377:   * returns true if motor is at reast
378:   */
379:  int Motor::is_stopped( )
380:  {
381:     return motor->is_stopped();
382:  }
383:
384:
385:  /**
386:   * returns true if the motor has been reversed internally
387:   */
388:  int Motor::is_reversed( )
389:  {
390:     return motor->is_reversed();
391:  }
392:
393:
394:
395:
396:
397:  //setter functions
398:
399:  /**
400:   * aquires lock and creates a new motor on a different port
401:   * exception safe to always release lock
402:   */
403:  int Motor::set_port( int port )
404:  {
405:     pros::motor_gearset_e_t gearset = motor->get_gearing();
406:     bool reversed = motor->is_reversed();
407:
408:     while ( lock.exchange( true ) );
409:
410:     try
411:     {
412:        delete motor;
413:        motor = new pros::Motor(port, gearset, reversed, pros::E_MOTOR_ENCODER_DEGREES);
414:        motor_port = port;
415:     }
416:     catch(...) //ensure lock will be released
417:     {
418:        Logger logger;
419:        log_entry entry;
420:        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not set port on motor port " + std::to_string(motor_port);
421:        entry.stream = "cerr";
422:        logger.add(entry);
423:
424:        lock.exchange(false);
425:        return 0;
426:     }
427:
428:     lock.exchange(false);
429:     return 1;
430:  }
431:
432:
433:  /**
434:   * aquires lock and sets zero position of motor
435:   * exception safe to always release lock
436:   */
437:  int Motor::tare_encoder( )
438:  {
439:     while ( lock.exchange( true ) );
440:
441:     try
442:     {
443:        motor->tare_position();
444:     }
445:     catch(...) //ensure lock will be released
446:     {
447:        Logger logger;
448:        log_entry entry;
449:        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not tare encoder on motor port " + std::to_string(motor_port);
450:        entry.stream = "cerr";
451:        logger.add(entry);
452:
```

```
453:         lock.exchange(false);
454:         return 0;
455:     }
456:
457:     lock.exchange(false);
458:
459:     return 1;
460: }
461:
462:
463: /**
464:  * aquires lock and sets new brake mode for motor
465:  * exception safe to always release lock
466:  */
467: int Motor::set_brake_mode( pros::motor_brake_mode_e_t brake_mode )
468: {
469:     while ( lock.exchange( true ) );
470:
471:     try
472:     {
473:         motor->set_brake_mode(brake_mode);
474:     }
475:     catch(...) //ensure lock will be released
476:     {
477:         Logger logger;
478:         log_entry entry;
479:         entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not set brakemode on motor port " + std::to_string(motor_port);
480:         entry.stream = "cerr";
481:         logger.add(entry);
482:
483:         lock.exchange(false);
484:         return 0;
485:     }
486:
487:     lock.exchange(false);
488:
489:     return 1;
490: }
491:
492:
493: /**
494:  * aquires lock and sets new gearing for motor
495:  * exception safe to always release lock
496:  */
497: int Motor::set_gearing( pros::motor_gearset_e_t gearset )
498: {
499:     while ( lock.exchange( true ) );
500:
501:     try
502:     {
503:         motor->set_gearing(gearset);
504:     }
505:     catch(...) //ensure lock will be released
506:     {
507:         Logger logger;
508:         log_entry entry;
509:         entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not set gearing on motor port " + std::to_string(motor_port);
510:         entry.stream = "cerr";
511:         logger.add(entry);
512:
513:         lock.exchange(false);
514:         return 0;
515:     }
516:
517:     lock.exchange(false);
518:
519:     return 1;
520: }
521:
522:
523: /**
524:  * aquires lock and internally reverses motor
525:  * exception safe to always release lock
526:  */
527: int Motor::reverse_motor( )
528: {
529:     while ( lock.exchange( true ) );
530:
531:     try
532:     {
533:         motor->set_reversed(!motor->is_reversed());
534:     }
535:     catch(...) //ensure lock will be released
536:     {
537:         Logger logger;
538:         log_entry entry;
539:         entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not reverse motor on port " + std::to_string(motor_port);
540:         entry.stream = "cerr";
541:         logger.add(entry);
542:
543:         lock.exchange(false);
544:         return 0;
545:     }
546:
547:     lock.exchange(false);
548:
549:     return 1;
550: }
551:
552:
553: /**
554:  * aquires lock and sets new PID constants for the motor
555:  * exception safe to always release lock
556:  */
557: int Motor::set_pid( pid pid_consts )
558: {
559:     while ( lock.exchange( true ) );
560:
561:     try
562:     {
563:         internal_motor_pid.kP = pid_consts.kP;
564:         internal_motor_pid.kI = pid_consts.kI;
565:         internal_motor_pid.kD = pid_consts.kD;
```

```
566:         internal_motor_pid.I_max = pid_consts.I_max;
567:     }
568:     catch(...) //ensure lock will be released
569:     {
570:         Logger logger;
571:         log_entry entry;
572:         entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not set motor pid on motor port " + std::to_string(motor_port);
573:         entry.stream = "cerr";
574:         logger.add(entry);
575:
576:         lock.exchange(false);
577:         return 0;
578:     }
579:
580:     lock.exchange(false);
581:
582:     return 1;
583: }
584:
585:
586: /**
587:  * sets a new log level for the motor, caps it between 0 and 5
588:  */
589: void Motor::set_log_level( int logging )
590: {
591:     if ( logging > 5 )
592:     {
593:         log_level = 5;
594:     }
595:     else if ( logging < 0 )
596:     {
597:         log_level = 0;
598:     }
599:     else
600:     {
601:         log_level = logging;
602:     }
603: }
604:
605:
606:
607:
608: //movement functions
609:
610: /**
611:  * sets new voltage by scaling from interval +/- 127 to +/- 12000
612:  */
613: int Motor::move( int voltage )
614: {
615:     int prev_max = 127;
616:     int prev_min = -127;
617:     int new_max = 12000;
618:     int new_min = -12000;
619:
620:     int scaled_voltage = (((voltage - prev_min) * (new_max - new_min)) / (prev_max - prev_min)) + new_min;
621:     set_voltage_setpoint(scaled_voltage); //dont aquire lock because it will be acquired in this function
622:     set_velocity_setpoint(to_velocity(scaled_voltage));
623:
624:     return 1;
625: }
626:
627: int Motor::user_move( int voltage ) {
628:     if(allow_driver_control) {
629:         move(voltage);
630:         return 1;
631:     }
632:
633:     return 0;
634: }
635:
636:
637: /**
638:  * sets new voltage by scaling from gearset interval to voltage range
639:  * of +/- 12000
640:  */
641: int Motor::move_velocity( int velocity )
642: {
643:     set_velocity_setpoint(velocity);
644:     set_voltage_setpoint(to_voltage(velocity));
645:
646:     return 1;
647: }
648:
649:
650: int Motor::set_voltage(int voltage) {
651:     set_voltage_setpoint(voltage);
652:     set_velocity_setpoint(to_velocity(voltage));
653:
654:     return 1;
655: }
656:
657:
658: /**
659:  * aquires lock and sets new voltage setpoint for the motor
660:  * exception safe to always release lock
661:  */
662: int Motor::set_voltage_setpoint( int voltage )
663: {
664:     while ( lock.exchange( true ) );
665:     voltage_setpoint = voltage;
666:     if ( voltage_setpoint != prev_voltage_setpoint )  //reset integral for new setpoint
667:     {
668:         integral = 0;
669:     }
670:     lock.exchange(false);
671:
672:     return 1;
673: }
674:
675:
676: int Motor::set_velocity_setpoint(int new_velocity) {
677:     while ( lock.exchange( true ) );
678:     velocity_setpoint = new_velocity;
```

```
679:     lock.exchange(false);
680:
681:     return 1;
682:   }
683:
684:
685:
686:
687:   //velocity pid control functions
688:
689:   /**
690:    * aquires lock and sets flag for using velocity PID
691:    */
692:   void Motor::set_motor_mode(motor_mode new_mode)
693:   {
694:     while ( lock.exchange( true ) );
695:     mode = new_mode;
696:     lock.exchange(false);
697:   }
698:
699:
700:
701:   //slew control functions
702:
703:   /**
704:    * aquires lock and sets new slew rate to be used in calculations
705:    */
706:   int Motor::set_slew( int rate )
707:   {
708:     while ( lock.exchange( true ) );
709:     slew_rate = rate;
710:     lock.exchange(false);
711:
712:     return 1;
713:   }
714:
715:
716:   /**
717:    * aquires lock and sets flag for using slew rate
718:    */
719:   void Motor::enable_slew( )
720:   {
721:     while ( lock.exchange( true ) );
722:     slew_enabled = true;
723:     lock.exchange(false);
724:   }
725:
726:
727:   /**
728:    * aquires lock and clears flag for using slew rate
729:    */
730:   void Motor::disable_slew( )
731:   {
732:     while ( lock.exchange( true ) );
733:     slew_enabled = false;
734:     lock.exchange(false);
735:   }
736:
737:
738:
739:
740:   //driver control lock setting and clearing functions
741:
742:   /**
743:    * aquires lock and sets flag for allowing driver control
744:    */
745:   void Motor::enable_driver_control()
746:   {
747:     while ( lock.exchange( true ) );
748:     allow_driver_control = true;
749:     lock.exchange(false);
750:   }
751:
752:
753:   /**
754:    * aquires lock and clears flag for allowing driver control
755:    */
756:   void Motor::disable_driver_control()
757:   {
758:     while ( lock.exchange( true ) );
759:     allow_driver_control = false;
760:     lock.exchange(false);
761:   }
762:
763:
764:   /**
765:    * returns flag for allowing driver control
766:    */
767:   int Motor::driver_control_allowed()
768:   {
769:     if ( allow_driver_control )
770:     {
771:       return 1;
772:     }
773:     else
774:     {
775:       return 0;
776:     }
777:   }
778:
779:
780:
781:
782:   /**
783:    * gets the voltage to set the motor to based on pid and slew rate calculations
784:    * and internally sets motor voltage
785:    * calculates what log message is to be based on the log level set and adds it to
786:    * the logger queue
787:    */
788:   int Motor::run( int delta_t )
789:   {
790:     switch(mode) {
791:       case e_builtin_velocity_pid: {
```

```
792:            motor->move_velocity(velocity_setpoint);
793:            break;
794:        } case e_voltage: {
795:            motor->move_voltage(voltage_setpoint);
796:            break;
797:        } case e_custom_velocity_pid: {
798:            int voltage = get_target_voltage( delta_t );
799:            motor->move_voltage(voltage);
800:            break;
801:        }
802:    }
803:
804:
805:
806:    std::string log_msg;
807:    switch ( log_level )
808:    {
809:        case 0:
810:            log_msg = "";
811:            break;
812:
813:        case 1:
814:            log_msg = (
815:                "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
816:                + ", Actual_Vol: " + std::to_string(get_actual_voltage())
817:                + ", Brake: " + std::to_string(get_brake_mode())
818:                + ", Gear: " + std::to_string(get_gearset())
819:                + ", I_max: " + std::to_string(internal_motor_pid.I_max)
820:                + ", I: " + std::to_string(integral)
821:                + ", kD: " + std::to_string(internal_motor_pid.kD)
822:                + ", kI: " + std::to_string(internal_motor_pid.kI)
823:                + ", kP: " + std::to_string(internal_motor_pid.kP)
824:                + ", Slew: " + std::to_string(get_slew_rate())
825:                + ", Time: " + std::to_string(pros::millis())
826:                + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
827:                + ", Vel: " + std::to_string(get_actual_velocity())
828:            );
829:            break;
830:
831:        case 2:
832:            log_msg = (
833:                "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
834:                + ", Actual_Vol: " + std::to_string(get_actual_voltage())
835:                + ", Brake: " + std::to_string(get_brake_mode())
836:                // + ", Calc_Target_Vol: " + std::to_string(voltage)
837:                + ", Gear: " + std::to_string(get_gearset())
838:                + ", I_max: " + std::to_string(internal_motor_pid.I_max)
839:                + ", I: " + std::to_string(integral)
840:                + ", kD: " + std::to_string(internal_motor_pid.kD)
841:                + ", kI: " + std::to_string(internal_motor_pid.kI)
842:                + ", kP: " + std::to_string(internal_motor_pid.kP)
843:                + ", Slew: " + std::to_string(get_slew_rate())
844:                + ", Target_Vol: " + std::to_string(voltage_setpoint)
845:                + ", Time: " + std::to_string(pros::millis())
846:                + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
847:                + ", Vel: " + std::to_string(get_actual_velocity())
848:            );
849:            break;
850:
851:        case 3:
852:            log_msg = (
853:                "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
854:                + ", Actual_Vol: " + std::to_string(get_actual_voltage())
855:                + ", Brake: " + std::to_string(get_brake_mode())
856:                // + ", Calc_Target_Vol: " + std::to_string(voltage)
857:                + ", Gear: " + std::to_string(get_gearset())
858:                + ", I_max: " + std::to_string(internal_motor_pid.I_max)
859:                + ", I: " + std::to_string(integral)
860:                + ", IME: " + std::to_string(get_encoder_position())
861:                + ", kD: " + std::to_string(internal_motor_pid.kD)
862:                + ", kI: " + std::to_string(internal_motor_pid.kI)
863:                + ", kP: " + std::to_string(internal_motor_pid.kP)
864:                + ", Slew: " + std::to_string(get_slew_rate())
865:                + ", Target_Vol: " + std::to_string(voltage_setpoint)
866:                + ", Time: " + std::to_string(pros::millis())
867:                + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
868:                + ", Vel: " + std::to_string(get_actual_velocity())
869:            );
870:            break;
871:
872:        case 4:
873:            log_msg = (
874:                "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
875:                + ", Actual_Vol: " + std::to_string(get_actual_voltage())
876:                + ", Brake: " + std::to_string(get_brake_mode())
877:                // + ", Calc_Target_Vol: " + std::to_string(voltage)
878:                + ", Dir: " + std::to_string(get_direction())
879:                + ", Gear: " + std::to_string(get_gearset())
880:                + ", I_max: " + std::to_string(internal_motor_pid.I_max)
881:                + ", I: " + std::to_string(integral)
882:                + ", IME: " + std::to_string(get_encoder_position())
883:                + ", kD: " + std::to_string(internal_motor_pid.kD)
884:                + ", kI: " + std::to_string(internal_motor_pid.kI)
885:                + ", kP: " + std::to_string(internal_motor_pid.kP)
886:                + ", Reversed: " + std::to_string(is_reversed())
887:                + ", Slew: " + std::to_string(get_slew_rate())
888:                + ", Target_Vol: " + std::to_string(voltage_setpoint)
889:                + ", Time: " + std::to_string(pros::millis())
890:                + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
891:                + ", Vel: " + std::to_string(get_actual_velocity())
892:            );
893:            break;
894:
895:        case 5:
896:            log_msg = (
897:                "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
898:                + ", Actual_Vol: " + std::to_string(get_actual_voltage())
899:                + ", Brake: " + std::to_string(get_brake_mode())
900:                // + ", Calc_Target_Vol: " + std::to_string(voltage)
901:                + ", Current: " + std::to_string(get_current_draw())
902:                + ", Dir: " + std::to_string(get_direction())
903:                + ", Gear: " + std::to_string(get_gearset())
904:                + ", I_max: " + std::to_string(internal_motor_pid.I_max)
```

```
905:                + ", I: " + std::to_string(integral)
906:                + ", IME: " + std::to_string(get_encoder_position())
907:                + ", kD: " + std::to_string(internal_motor_pid.kD)
908:                + ", kI: " + std::to_string(internal_motor_pid.kI)
909:                + ", kP: " + std::to_string(internal_motor_pid.kP)
910:                + ", Reversed: " + std::to_string(is_reversed())
911:                + ", Slew: " + std::to_string(get_slew_rate())
912:                + ", Target_Vol: " + std::to_string(voltage_setpoint)
913:                + ", Temp: " + std::to_string(get_temperature())
914:                + ", Time: " + std::to_string(pros::millis())
915:                + ", Torque: " + std::to_string(get_torque())
916:                + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
917:                + ", Vel: " + std::to_string(get_actual_velocity())
918:            );
919:            break;
920:
921:    }
922:
923:    Logger logger;
924:    log_entry entry;
925:    entry.content = log_msg;
926:    entry.stream = "clog";
927:    logger.add(entry);
928:
929:    return 1;
930: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/motors/MotorThread.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 2/16/2020
5:   * @reviewed_by: Aiden Carney
6:   * TODO:
7:   *
8:   * contains functions that handle motor functions
9:   */
10:
11: #ifndef __MOTORTHREAD_HPP__
12: #define __MOTORTHREAD_HPP__
13:
14: #include <vector>
15: #include <atomic>
16:
17: #include "main.h"
18:
19: #include "../../Configuration.hpp"
20: #include "Motor.hpp"
21:
22:
23: /**
24:  * @see: Motor.hpp
25:  *
26:  * contains singleton class for using motors in a thread
27:  * motors are added to a vector and iterated over in a thread so that the voltage
28:  * can be set
29:  */
30: class MotorThread
31: {
32:     private:
33:         MotorThread();
34:         static MotorThread *thread_obj;
35:
36:         static std::vector<Motor*> motors;
37:         static std::atomic<bool> lock;  //protect vector from concurrent access
38:
39:
40:         /**
41:          * @param: void* -> not used, but necessary to follow thread making constructor
42:          * @return: None
43:          *
44:          * the function to be run on a thread that calls the run function for
45:          * each motor that sets the voltage and performs logging
46:          */
47:         static void run(void*);
48:
49:         pros::Task *thread;  // the motor thread
50:
51:
52:     public:
53:         ~MotorThread();
54:
55:         /**
56:          * @return: MotorThread -> instance of class to be used throughout program
57:          *
58:          * give the instance of the singleton class or creates it if it does
59:          * not yet exist
60:          */
61:         static MotorThread* get_instance();
62:
63:
64:
65:
66:         /**
67:          * @return: None
68:          *
69:          * starts the thread or resmes it if it was stopped
70:          */
71:         void start_thread();
72:
73:         /**
74:          * @return: None
75:          *
76:          * stops the thread from being scheduled
77:          */
78:         void stop_thread();
79:
80:
81:
82:
83:         /**
84:          * @param: Motor &motor -> the motor to add to the vector
85:          * @return: int -> 1 if motor was successfully added, 0 otherwise
86:          *
87:          * adds a motor to the vector of motors to operate
88:          * logs that the motor was added to the logger queue
89:          */
90:         int register_motor( Motor &motor );
91:
92:         /**
93:          * @param: Motor &motor -> the motor to remove from the vector
94:          * @return: int -> 1 if motor was successfully added, 0 otherwise
95:          *
96:          * removes a motor from the vector of motors to operate
97:          * logs that the motor was removed to the logger queue
98:          */
99:         int unregister_motor( Motor &motor );
100:
101:        int is_registered(Motor &motor);
102:
103:
104: };
105:
106: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/motors/MotorThread.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains implementation for functions that handle motor functions
8:   */
9:
10:  #include <atomic>
11:  #include <stdio.h>
12:  #include <vector>
13:
14:  #include "main.h"
15:
16:  #include "../serial/Logger.hpp"
17:  #include "Motor.hpp"
18:  #include "MotorThread.hpp"
19:
20:
21:  MotorThread *MotorThread::thread_obj = NULL;
22:  std::vector<Motor*> MotorThread::motors;
23:  std::atomic<bool> MotorThread::lock = ATOMIC_VAR_INIT(false);
24:
25:
26:  MotorThread::MotorThread()
27:  {
28:      thread = new pros::Task( run, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "motor_thread");
29:      thread->suspend();
30:  }
31:
32:
33:  MotorThread::~MotorThread()
34:  {
35:      thread->remove();
36:      delete thread;
37:  }
38:
39:
40:  void MotorThread::run(void*)
41:  {
42:      int start = pros::millis();
43:      while (1) {
44:          while ( lock.exchange( true ) );
45:          for ( int i = 0; i < motors.size(); i++ ) {
46:              motors.at(i)->run( pros::millis() - start );
47:          }
48:          start = pros::millis();
49:          lock.exchange(false);
50:          pros::delay(5);
51:      }
52:  }
53:
54:
55:
56:  /**
57:   * inits object if object is not already initialized based on a static bool
58:   * sets bool if it is not set
59:   */
60:  MotorThread* MotorThread::get_instance() {
61:      if ( thread_obj == NULL ) {
62:          thread_obj = new MotorThread;
63:      }
64:      return thread_obj;
65:  }
66:
67:
68:  void MotorThread::start_thread() {
69:      thread->resume();
70:  }
71:
72:  void MotorThread::stop_thread() {
73:      thread->suspend();
74:  }
75:
76:
77:  int MotorThread::register_motor( Motor &motor ) {
78:      while ( lock.exchange( true ) );
79:
80:      Logger logger;
81:      log_entry entry;
82:      char buffer[10];
83:
84:
85:      try
86:      {
87:          motors.push_back(&motor);
88:
89:          sprintf(buffer, "%p", &motor);
90:          entry.stream = "clog";
91:          entry.content = "[INFO], " + std::to_string(pros::millis()) +  ", motor added at " + buffer;
92:          logger.add(entry);
93:      }
94:      catch ( ... )
95:      {
96:          sprintf(buffer, "%p", &motor);
97:          entry.content = "[WARNING], " + std::to_string(pros::millis()) +  ", could not add motor at " + buffer;
98:          entry.stream = "cerr";
99:          logger.add(entry);
100:
101:         lock.exchange(false);
102:         return 0;
103:     }
104:
105:     lock.exchange(false);
106:     return 1;
107: }
108:
109:
110: int MotorThread::unregister_motor( Motor &motor )
111: {
112:     while ( lock.exchange( true ) );
113:
```

```
114:     Logger logger;
115:     log_entry entry;
116:     char buffer[10];
117:
118:     auto element = std::find(begin(motors), end(motors), &motor);
119:     if ( element != motors.end())
120:     {
121:         motors.erase(element);
122:
123:         sprintf(buffer, "%p", &motor);
124:         entry.stream = "clog";
125:         entry.content = "[INFO] " + std::to_string(pros::millis()) + ", motor removed at " + buffer;
126:         logger.add(entry);
127:     }
128:     else
129:     {
130:         sprintf(buffer, "%p", &motor);
131:         entry.content = "[WARNING] " + std::to_string(pros::millis()) +  ", could not remove motor at " + buffer;
132:         entry.stream = "cerr";
133:         logger.add(entry);
134:
135:         lock.exchange(false);
136:         return 0;
137:     }
138:
139:     lock.exchange(false);
140:     return 1;
141:
142: }
143:
144:
145: int MotorThread::is_registered(Motor &motor) {
146:     int registered = 0;
147:
148:     while ( lock.exchange( true ) );
149:
150:     auto element = std::find(begin(motors), end(motors), &motor);
151:     if ( element != motors.end()) {
152:         registered = 1;
153:     }
154:
155:     lock.exchange(false);
156:
157:     return registered;
158: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/motors/Motors.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 2/16/2020
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains global struct for all motors
8:   */
9:
10: #ifndef __MOTORS_HPP__
11: #define __MOTORS_HPP__
12:
13: #include <array>
14:
15: #include "main.h"
16:
17: #include "../../Configuration.hpp"
18: #include "Motor.hpp"
19:
20:
21: namespace Motors
22: {
23:     extern Motor front_right;
24:     extern Motor front_left;
25:     extern Motor back_right;
26:     extern Motor back_left;
27:     extern Motor left_intake;
28:     extern Motor right_intake;
29:     extern Motor upper_indexer;
30:     extern Motor lower_indexer;
31:
32:     extern double chassis_gear_ratio;
33:
34:     extern std::array<Motor*, 8> motor_array;
35:     extern std::array<std::string, 8> motor_names_array;
36:
37:     void enable_driver_control();
38:     void disable_driver_control();
39:     void set_brake_mode(pros::motor_brake_mode_e_t new_brakemode);
40:     void stop_all_motors();
41:     void set_log_level(int log_level);
42:     void register_motors();
43:     void unregister_motors();
44: };
45:
46:
47: #endif
```

```cpp
/**
 * @file: ./RobotCode/src/motors/Motors.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * contains definition of global struct
 */

#include "Motors.hpp"
#include "MotorThread.hpp"

namespace Motors
{
    Motor front_right {Configuration::get_instance()->front_right_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->front_right_reversed};
    Motor front_left {Configuration::get_instance()->front_left_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->front_left_reversed};
    Motor back_right {Configuration::get_instance()->back_right_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->back_right_reversed};
    Motor back_left {Configuration::get_instance()->back_left_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->back_left_reversed};
    Motor left_intake {Configuration::get_instance()->left_intake_port, pros::E_MOTOR_GEARSET_36, Configuration::get_instance()->left_intake_reversed};
    Motor right_intake {Configuration::get_instance()->right_intake_port, pros::E_MOTOR_GEARSET_36, Configuration::get_instance()->right_intake_reversed};
    Motor upper_indexer {Configuration::get_instance()->upper_indexer_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->upper_indexer_reversed};
    Motor lower_indexer {Configuration::get_instance()->lower_indexer_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->lower_indexer_reversed};

    double chassis_gear_ratio = 3 / 5;

    std::array<Motor*, 8> motor_array = {
        &front_right,
        &front_left,
        &back_right,
        &back_left,
        &left_intake,
        &right_intake,
        &upper_indexer,
        &lower_indexer,
    };

    std::array<std::string, 8> motor_names_array = {
        "Front Left",
        "Front Right",
        "Back Left",
        "Back Right",
        "Left Intake",
        "Right Intake",
        "Upper Indexer",
        "Lower Indexer"
    };

    void enable_driver_control() {
        Motors::front_left.enable_driver_control();
        Motors::front_left.enable_driver_control();
        Motors::back_right.enable_driver_control();
        Motors::back_left.enable_driver_control();
        Motors::left_intake.enable_driver_control();
        Motors::right_intake.enable_driver_control();
        Motors::upper_indexer.enable_driver_control();
        Motors::lower_indexer.enable_driver_control();
    }

    void disable_driver_control() {
        Motors::front_left.disable_driver_control();
        Motors::front_left.disable_driver_control();
        Motors::back_right.disable_driver_control();
        Motors::back_left.disable_driver_control();
        Motors::left_intake.disable_driver_control();
        Motors::right_intake.disable_driver_control();
        Motors::upper_indexer.disable_driver_control();
        Motors::lower_indexer.disable_driver_control();
    }

    void set_brake_mode(pros::motor_brake_mode_e_t new_brakemode) {
        Motors::front_left.set_brake_mode(new_brakemode);
        Motors::front_left.set_brake_mode(new_brakemode);
        Motors::back_right.set_brake_mode(new_brakemode);
        Motors::back_left.set_brake_mode(new_brakemode);
        Motors::left_intake.set_brake_mode(new_brakemode);
        Motors::right_intake.set_brake_mode(new_brakemode);
        Motors::upper_indexer.set_brake_mode(new_brakemode);
        Motors::lower_indexer.set_brake_mode(new_brakemode);
    }

    void stop_all_motors() {
        Motors::front_left.move(0);
        Motors::front_left.move(0);
        Motors::back_right.move(0);
        Motors::back_left.move(0);
        Motors::left_intake.move(0);
        Motors::right_intake.move(0);
        Motors::upper_indexer.move(0);
        Motors::lower_indexer.move(0);
    }

    void set_log_level(int log_level) {
        Motors::front_right.set_log_level(log_level);
        Motors::front_left.set_log_level(log_level);
        Motors::back_right.set_log_level(log_level);
        Motors::back_left.set_log_level(log_level);
        Motors::left_intake.set_log_level(log_level);
        Motors::right_intake.set_log_level(log_level);
        Motors::upper_indexer.set_log_level(log_level);
        Motors::lower_indexer.set_log_level(log_level);
    }

    void register_motors() {
        MotorThread* motor_thread = MotorThread::get_instance();
        motor_thread->register_motor(Motors::front_right);
        motor_thread->register_motor(Motors::front_left);
        motor_thread->register_motor(Motors::back_right);
        motor_thread->register_motor(Motors::back_left);
        motor_thread->register_motor(Motors::left_intake);
        motor_thread->register_motor(Motors::right_intake);
        motor_thread->register_motor(Motors::upper_indexer);
        motor_thread->register_motor(Motors::lower_indexer);
    }
```

```
114:
115:    void unregister_motors() {
116:        MotorThread* motor_thread = MotorThread::get_instance();
117:        motor_thread->unregister_motor(Motors::front_right);
118:        motor_thread->unregister_motor(Motors::front_left);
119:        motor_thread->unregister_motor(Motors::back_right);
120:        motor_thread->unregister_motor(Motors::back_left);
121:        motor_thread->unregister_motor(Motors::left_intake);
122:        motor_thread->unregister_motor(Motors::right_intake);
123:        motor_thread->unregister_motor(Motors::upper_indexer);
124:        motor_thread->unregister_motor(Motors::lower_indexer);
125:    }
126:
127: };
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/sensors/AnalogInSensor.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains a wrapper class for ADI analog in sensor
8:   */
9:
10: #ifndef __ANALOGINSENSOR_HPP__
11: #define __ANALOGINSENSOR_HPP__
12:
13: #include <atomic>
14: #include <vector>
15:
16: #include "main.h"
17:
18:
19: class AnalogInSensor
20: {
21:     private:
22:         pros::ADIAnalogIn *sensor;
23:         bool calibrated;
24:
25:     public:
26:         AnalogInSensor();
27:         AnalogInSensor(char port);
28:         AnalogInSensor(pros::ext_adi_port_pair_t port_pair);
29:         ~AnalogInSensor();
30:
31:         void set_port(char port);
32:         void set_port(pros::ext_adi_port_pair_t port_pair);
33:
34:         double get_raw_value();
35:         double get_value(bool high_res);
36:
37:         void calibrate();
38:         bool is_calibrated();
39: };
40:
41:
42:
43:
44:
45: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/sensors/AnalogInSensor.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains implementation for wrapper class for analog in sensor
8:   */
9:
10: #include "../serial/Logger.hpp"
11: #include "AnalogInSensor.hpp"
12:
13:
14:
15: AnalogInSensor::AnalogInSensor() {
16:     sensor = NULL;
17: }
18:
19: AnalogInSensor::AnalogInSensor(char port) {
20:     sensor = new pros::ADIAnalogIn(port);
21: }
22:
23: AnalogInSensor::AnalogInSensor(pros::ext_adi_port_pair_t port_pair) {
24:     sensor = new pros::ADIAnalogIn(port_pair);
25: }
26:
27: AnalogInSensor::~AnalogInSensor()
28: {
29:     if(sensor != NULL) {
30:         delete sensor;
31:     }
32: }
33:
34:
35: void AnalogInSensor::set_port(char port) {
36:     if(sensor != NULL) {
37:         delete sensor;
38:     }
39:
40:     sensor = new pros::ADIAnalogIn(port);
41: }
42:
43:
44: void AnalogInSensor::set_port(pros::ext_adi_port_pair_t port_pair) {
45:     if(sensor != NULL) {
46:         delete sensor;
47:     }
48:
49:     sensor = new pros::ADIAnalogIn(port_pair);
50: }
51:
52:
53: double AnalogInSensor::get_raw_value() {
54:     double value = sensor->get_value();
55:     return value;
56:
57: }
58:
59:
60: double AnalogInSensor::get_value(bool high_res) {
61:     if(!calibrated)
62:     {
63:         Logger logger;
64:         log_entry entry;
65:         entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not read analog sensor (not calibrated) ";
66:         entry.stream = "cerr";
67:
68:         logger.add(entry);
69:
70:         return INT32_MAX;
71:     }
72:
73:     if(high_res)
74:     {
75:         return sensor->get_value_calibrated_HR();
76:     }
77:     else
78:     {
79:         return sensor->get_value_calibrated();
80:     }
81: }
82:
83:
84: void AnalogInSensor::calibrate() {
85:     sensor->calibrate();
86:     calibrated = true;
87:
88: }
89:
90:
91: bool AnalogInSensor::is_calibrated() {
92:     return calibrated;
93: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/sensors/BallDetector.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains a wrapper class for the encoders
8:   */
9:
10: #ifndef __BALLDETECTOR_HPP__
11: #define __BALLDETECTOR_HPP__
12:
13: #include <tuple>
14:
15: #include "main.h"
16:
17: #include "AnalogInSensor.hpp"
18:
19:
20: class BallDetector
21: {
22:     private:
23:         AnalogInSensor ball_detector_top;
24:         AnalogInSensor ball_detector_filter;
25:         AnalogInSensor ball_detector_bottom;
26:         pros::Optical* optical_sensor;
27:
28:         int time_since_last_ball;
29:         bool log_data;
30:
31:         int threshold;
32:
33:     public:
34:         BallDetector(
35:             AnalogInSensor& detector_top_left,
36:             AnalogInSensor& detector_filter,
37:             AnalogInSensor& detector_bottom,
38:             int optical_port,
39:             int detector_threshold
40:         );
41:         ~BallDetector();
42:
43:         int set_threshold(int new_threshold);
44:         int check_filter_level();
45:         std::vector<bool> locate_balls();
46:
47:         void set_led_brightness(int pct);
48:         void auto_set_led_brightness();
49:
50:         std::tuple<int, int> debug_color();
51:         void start_logging();
52:         void stop_logging();
53:
54: };
55:
56:
57:
58:
59:
60: #endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/sensors/BallDetector.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains implementation for ball detector class
 */

#include "../serial/Logger.hpp"
#include "BallDetector.hpp"



BallDetector::BallDetector(
    AnalogInSensor& detector_top_left,
    AnalogInSensor& detector_filter,
    AnalogInSensor& detector_bottom,
    int optical_port,
    int detector_threshold
) {
    ball_detector_top = detector_top_left;
    ball_detector_filter = detector_filter;
    ball_detector_bottom = detector_bottom;
    optical_sensor = new pros::Optical(optical_port);

    optical_sensor->disable_gesture();
    optical_sensor->set_led_pwm(50);

    threshold = detector_threshold;
    time_since_last_ball = 0;
    log_data = false;
}
BallDetector::~BallDetector() {
    delete optical_sensor;
}


int BallDetector::set_threshold(int new_threshold) {
    threshold = new_threshold;
    return 1;
}


int BallDetector::check_filter_level() {
    int return_code = 0;
    if(ball_detector_filter.get_raw_value() < threshold) {  // ball is detected
        time_since_last_ball = 0;  // ball detected so there is no time since last ball

        double hue = optical_sensor->get_hue();
        if(hue > 170 && hue < 260) {  // color is blue
            return_code = 1;
        } else if(hue > 335 || hue < 25) {  // color is red
            return_code = 2;
        } else {  // could not determine color based on ranges
            return_code = -1;
        }
    } else {
        time_since_last_ball = pros::millis() - time_since_last_ball;  // get time elapsed
    }

    if(log_data) {
        Logger logger;
        log_entry entry;
        entry.content = (
            "[INFO] " + std::string("BALL_DETECT_MIDDLE")
            + ", Time: " + std::to_string(pros::millis())
            + ", ball_detected: " + std::to_string(return_code)
            + ", time_since_last_ball " + std::to_string(time_since_last_ball)
            + ", line_detector: " + std::to_string(ball_detector_filter.get_raw_value())
            + ", threshold: " + std::to_string(threshold)

        );
        entry.stream = "clog";
        logger.add(entry);
    }


    return return_code;  // no ball is detected
}


std::vector<bool> BallDetector::locate_balls() {
    std::vector<bool> locations;
    if(ball_detector_top.get_raw_value() < threshold) {
        locations.push_back(true);
    } else {
        locations.push_back(false);
    }

    if(ball_detector_filter.get_raw_value() < threshold) {
        locations.push_back(true);
    } else {
        locations.push_back(false);
    }

    if(ball_detector_bottom.get_raw_value() < threshold) {
        locations.push_back(true);
    } else {
        locations.push_back(false);
    }

    if(log_data) {
        Logger logger;
        log_entry entry;
        entry.content = (
            "[INFO] " + std::string("BALL_DETECT_MIDDLE")
            + ", time: " + std::to_string(pros::millis())
            + ", top_present: " + std::to_string(locations.at(0))
            + ", middle_present: " + std::to_string(locations.at(1))
            + ", bottom_present: " + std::to_string(locations.at(2))
            + ", top: " + std::to_string(ball_detector_top.get_raw_value())
```

```
114:             + ", middle: " + std::to_string(ball_detector_filter.get_raw_value())
115:             + ", bottom: " + std::to_string(ball_detector_bottom.get_raw_value())
116:             + ", threshold: " + std::to_string(threshold)
117:         );
118:         entry.stream = "clog";
119:         logger.add(entry);
120:     }
121:
122:
123:     return locations;
124: }
125:
126:
127: void BallDetector::set_led_brightness(int pct) {
128:     optical_sensor->set_led_pwm(pct);
129: }
130:
131:
132: void BallDetector::auto_set_led_brightness() {
133:     int pct = 100 * std::abs(1 - optical_sensor->get_brightness());  // 1 to 1 scale
134:     optical_sensor->set_led_pwm(pct);
135: }
136:
137:
138: std::tuple<int, int> BallDetector::debug_color() {
139:     return std::make_tuple(ball_detector_filter.get_raw_value(), check_filter_level());
140: }
141:
142: void BallDetector::start_logging() {
143:     log_data = true;
144: }
145:
146: void BallDetector::stop_logging() {
147:     log_data = false;
148: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/sensors/Encoder.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains a wrapper class for the encoders
8:   */
9:
10: #ifndef __ENCODER_HPP__
11: #define __ENCODER_HPP__
12:
13: #include <atomic>
14: #include <unordered_map>
15:
16: #include "main.h"
17:
18:
19: class Encoder
20: {
21:     private:
22:         pros::ADIEncoder *encoder;
23:
24:         std::atomic<bool> lock;  // protect map from concurrent access
25:         int latest_uid;
26:         std::unordered_map<int, double> zero_positions;
27:
28:     public:
29:         Encoder(char upper_port, char lower_port, bool reverse);
30:         ~Encoder();
31:
32:         int get_unique_id(bool zero=false);
33:
34:         double get_position(int unique_id);
35:         double get_absolute_position(bool scaled);
36:
37:         int reset(int unique_id);
38:
39:         void forget_position(int unique_id);
40:
41: };
42:
43:
44:
45:
46:
47: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/sensors/Encoder.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains implementation for wrapper class for Encoder
8:   */
9:
10:  #include <atomic>
11:  #include <vector>
12:
13:  #include "main.h"
14:
15:  #include "../serial/Logger.hpp"
16:  #include "Encoder.hpp"
17:
18:
19:
20:  Encoder::Encoder( char upper_port, char lower_port, bool reverse )
21:  {
22:      lock = ATOMIC_VAR_INIT(false);
23:
24:      encoder = new pros::ADIEncoder(upper_port, lower_port, reverse);
25:
26:
27:      while ( lock.exchange( true ) ); //aquire lock
28:      latest_uid = 0;
29:      zero_positions[0] = encoder->get_value();
30:      lock.exchange(false);  //release lock
31:  }
32:
33:
34:  Encoder::~Encoder()
35:  {
36:      // TODO: figure out why checking for null pointer needs to be present to not crash when program starts
37:      if(encoder != NULL)  // causes segfault when program begins if this is not present
38:      {
39:          delete encoder;
40:      }
41:  }
42:
43:
44:
45:
46:  int Encoder::get_unique_id(bool zero /*false*/)
47:  {
48:      while ( lock.exchange( true ) ); //aquire lock
49:
50:      latest_uid += 1;
51:      int id = latest_uid;
52:      zero_positions[id] = zero_positions.at(0);
53:      lock.exchange(false);  //release lock
54:
55:      if(zero) {
56:          reset(id);
57:      }
58:
59:      return id;
60:  }
61:
62:
63:
64:
65:  double Encoder::get_position(int unique_id)
66:  {
67:      if(zero_positions.find(unique_id) == zero_positions.end())
68:      {
69:          Logger logger;
70:          log_entry entry;
71:          entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not get encoder position with unique id " + std::to_string(unique_id);
72:          entry.stream = "cerr";
73:
74:          logger.add(entry);
75:
76:          return INT32_MAX;
77:      }
78:
79:      double position = get_absolute_position(false) - zero_positions.at(unique_id);
80:      return position;
81:
82:  }
83:
84:
85:
86:  double Encoder::get_absolute_position(bool scaled)
87:  {
88:      double position = encoder->get_value() - zero_positions.at(0);
89:
90:      if(scaled)
91:      {
92:          position = ((int)position % 720) - 360;  // scales to interval [-360,360]
93:      }
94:
95:      return position;
96:  }
97:
98:
99:
100:
101: int Encoder::reset(int unique_id)
102: {
103:     if(zero_positions.find(unique_id) == zero_positions.end() || unique_id == 0)
104:     {
105:         Logger logger;
106:         log_entry entry;
107:         entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not get encoder position with unique id " + std::to_string(unique_id);
108:         entry.stream = "cerr";
109:
110:         logger.add(entry);
111:
112:         return 0;
113:     }
```

```
114:
115:    zero_positions.at(unique_id) = encoder->get_value();
116:    return 1;
117: }
118:
119:
120: void Encoder::forget_position(int unique_id) {
121:    if(zero_positions.find(unique_id) == zero_positions.end() || unique_id == 0)
122:    {
123:        Logger logger;
124:        log_entry entry;
125:        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not remove zero position with unique id " + std::to_string(unique_id);
126:        entry.stream = "cerr";
127:
128:        logger.add(entry);
129:
130:    }
131:    zero_positions.erase(unique_id);
132: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/sensors/Sensors.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 2/29/2020
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains a class for interacting with the ADI sensors on the robot
8:   */
9:
10: #ifndef __SENSORS_HPP__
11: #define __SENSORS_HPP__
12:
13: #include "main.h"
14:
15: #include "BallDetector.hpp"
16: #include "Encoder.hpp"
17: #include "AnalogInSensor.hpp"
18:
19:
20:
21: namespace Sensors
22: {
23:     extern Encoder right_encoder;
24:     extern Encoder left_encoder;
25:     extern Encoder strafe_encoder;
26:
27:     extern AnalogInSensor line_tracker_top;
28:     extern AnalogInSensor line_tracker_middle;
29:     extern AnalogInSensor line_tracker_bottom;
30:     extern pros::Optical optical;
31:     extern BallDetector ball_detector;
32:
33:     extern pros::Imu imu;
34:     extern bool imu_is_calibrated;
35:
36:     void calibrate_imu();
37:     void log_data();
38:     std::tuple<double, double> get_average_encoders(int l_id, int r_id);
39: }
40:
41:
42:
43:
44: #endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/sensors/Sensors.cpp
 * @author: Aiden Carney
 * @reviewed_on: 2/29/2020
 * @reviewed_by: Aiden Carney
 *
 * @see: Sensors.hpp
 *
 * contains definitions for sensors and implementation for sensor class
 */

#include "Sensors.hpp"
#include "../motors/Motors.hpp"
#include "../../Configuration.hpp"
#include "../serial/Logger.hpp"


namespace Sensors
{
    Encoder right_encoder{RIGHT_ENC_TOP_PORT, RIGHT_ENC_BOTTOM_PORT, false};
    Encoder left_encoder{LEFT_ENC_TOP_PORT, LEFT_ENC_BOTTOM_PORT, false};
    Encoder strafe_encoder{STRAFE_ENC_TOP_PORT, STRAFE_ENC_BOTTOM_PORT, true};

    AnalogInSensor line_tracker_top{DETECTOR_TOP_PORT};
    AnalogInSensor line_tracker_middle{DETECTOR_MIDDLE_PORT};
    AnalogInSensor line_tracker_bottom{DETECTOR_BOTTOM_PORT};

    // BallDetector ball_detector{DETECTOR_TOP_PORT, DETECTOR_MIDDLE_PORT, DETECTOR_BOTTOM_PORT, VISIONSENSOR_PORT, Configuration::get_instance()->filter_threshold};
    BallDetector ball_detector{
        line_tracker_top,
        line_tracker_middle,
        line_tracker_bottom,
        OPTICAL_PORT,
        Configuration::get_instance()->filter_threshold
    };

    pros::Imu imu{IMU_PORT};
    bool imu_is_calibrated = false;

    void calibrate_imu() {
        bool calibrated = false;
        while(!calibrated) {  // block until imu is connected and calibrated
            imu.reset();  // calibrate imu
            while(imu.is_calibrating()) {
                pros::delay(10);
                calibrated = true;
            }
        }
        imu_is_calibrated = true;
    }

    void log_data() {
        Logger logger;
        log_entry entry;
        entry.content = ("[INFO], " + std::to_string(pros::millis())
            + ", Sensor Data"
            +  ", Right_Enc: " + std::to_string(right_encoder.get_absolute_position(false))
            +  ", Left_Enc: " + std::to_string(left_encoder.get_absolute_position(false))
            +  ", Top Detector" + std::to_string(ball_detector.locate_balls().at(0))
            +  ", Middle Detector" + std::to_string(ball_detector.locate_balls().at(1))
            +  ", Bottom Detector" + std::to_string(ball_detector.locate_balls().at(2))
        );
        entry.stream = "clog";
        logger.add(entry);
    }

    /**
     * takes the average of each side of the drive encoders
     * hopefully to reduce error of encoders
     * returns tuple of encoder values
     */
    std::tuple<double, double> get_average_encoders(int l_id, int r_id) {
        // use a weighted average to merge all encoders on the robot for a hopefully more accurate reading
        double left_encoder_val = (0 * Motors::front_left.get_encoder_position() * Motors::chassis_gear_ratio) + (0 * Motors::back_left.get_encoder_position() * Motors::chassis_gear_ratio) + (1 * Sensors::left_encoder.get_position(l_id));
        double right_encoder_val = (0 * Motors::front_right.get_encoder_position() * Motors::chassis_gear_ratio) + (0 * Motors::back_right.get_encoder_position() * Motors::chassis_gear_ratio) + (1 * Sensors::right_encoder.get_position(r_id));

        return {left_encoder_val, right_encoder_val};
    }


}
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/lcdCode/Gimmicks.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   * TODO: fix loading screen, it sometimes does not work
7:   *
8:   * contains lcd gimmicks that are used to enhance interface
9:   *
10:  */
11:
12:  #ifndef __GIMMICKS_HPP__
13:  #define __GIMMICKS_HPP__
14:
15:  #include <string>
16:
17:  #include "../../../include/main.h"
18:
19:  #include "Styles.hpp"
20:
21:
22:  /**
23:   * @see: Styles.hpp
24:   * @see: ./lcdCode
25:   *
26:   * used to display warning box
27:   */
28:  class WarningMessage : virtual Styles
29:  {
30:      protected:
31:          /**
32:           * @param: lv_obj_t* mbox -> message box object
33:           * @param: const char* txt -> text for message box
34:           * @return: LV_RES_OK -> if finishes successfully
35:           *
36:           * @see: Styles.hpp
37:           * @see: ./lcdCode
38:           *
39:           * sets static int option to positive or negative based on feedback
40:           *
41:           */
42:          static lv_res_t mbox_apply_action(lv_obj_t * mbox, const char * txt);
43:          static const char* buttons[];
44:          static int option;
45:
46:          lv_obj_t *warn_box;
47:
48:
49:      public:
50:          WarningMessage();
51:          virtual ~WarningMessage();
52:
53:          /**
54:           * @param: std::string warn_msg -> message that will appear as option
55:           * @param: lv_obj_t* parent -> the parent that the message box will appear on
56:           * @return: bool -> if user selected yes or no
57:           *
58:           * returns true or false based on what user selects
59:           * implementation of this is up to user
60:           *
61:           */
62:          bool warn(std::string warn_msg, lv_obj_t *parent);
63:
64:  };
65:
66:
67:  /**
68:   * @see: Styles.hpp
69:   * @see: ./lcdCode
70:   *
71:   * methods and objects for a loading bar
72:   */
73:  class Loading : virtual Styles
74:  {
75:      protected:
76:          lv_obj_t *loader;
77:
78:      public:
79:          Loading();
80:          ~Loading();
81:
82:          /**
83:           * @param: int estimated_duration -> duration that loading should take used to set speed of bar
84:           * @param: lv_obj_t* parent -> parent object that loading bar will go on
85:           * @param: int x -> x position of loading bar relative to parent
86:           * @param int y -> y position of loading bar relative to parent
87:           * @return: None
88:           *
89:           * shows the loader and starts the action of it moving
90:           *
91:           */
92:          void show_load(int estimated_duration, lv_obj_t *parent, int x, int y); //starts the loader
93:
94:          /**
95:           * @return: None
96:           *
97:           * hides the loader
98:           * this should be about when the loader is finished
99:           * Used to keep a smooth transition
100:          *
101:          */
102:         void hide_load(); //ends the loader and hides it
103: };
104:
105:
106:
107:
108:
109:
110:
111: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/lcdCode/Gimmicks.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see: Gimmicks.hpp
8:   *
9:   * contains implementation for header file
10:  *
11:  */
12:
13: #include "../../../include/main.h"
14: #include "../../../include/api.h"
15:
16: #include "Styles.hpp"
17: #include "Gimmicks.hpp"
18:
19:
20:
21: const char* WarningMessage::buttons[] = {"Back", "Continue", ""};
22:
23: //base classes
24: int WarningMessage::option = 0;
25:
26: WarningMessage::WarningMessage()
27: {
28:     option = 0;
29:
30:     warn_box = lv_mbox_create(lv_scr_act(), NULL);
31:     lv_mbox_set_text(warn_box, "None");
32:     lv_mbox_add_btns(warn_box, buttons, NULL);
33:     lv_mbox_set_action(warn_box, mbox_apply_action);
34:
35:     lv_mbox_set_style(warn_box, LV_MBOX_STYLE_BG, &warn_box_bg);
36:     lv_mbox_set_style(warn_box, LV_MBOX_STYLE_BTN_REL, &warn_box_released);
37:     lv_mbox_set_style(warn_box, LV_MBOX_STYLE_BTN_PR, &warn_box_pressed);
38:
39:     lv_obj_set_width(warn_box, 400);
40:     lv_obj_set_height(warn_box, 140);
41:
42:     lv_obj_align(warn_box, NULL, LV_ALIGN_CENTER, 0, -50);
43:
44:
45: }
46:
47: WarningMessage::~WarningMessage()
48: {
49:     lv_obj_del(warn_box);
50: }
51:
52: /**
53:  * compares text of message to set option to positive or negative
54:  */
55: lv_res_t WarningMessage::mbox_apply_action(lv_obj_t * mbox, const char * txt)
56: {
57:     if ( txt == "Continue" )
58:     {
59:         option = 1;
60:     }
61:
62:     else if ( txt == "Back" )
63:     {
64:         option = -1;
65:     }
66:
67:     return LV_RES_OK;
68: }
69:
70: /**
71:  * dislays a message box and sets the text
72:  * user can choose "continue" or "back"
73:  * how function works line 3
74:  */
75: bool WarningMessage::warn( std::string warn_msg, lv_obj_t *parent )
76: {
77:     option = 0;
78:
79:     lv_obj_set_hidden(warn_box, false);
80:     lv_obj_set_parent(warn_box, parent);
81:     lv_mbox_set_text(warn_box, warn_msg.c_str());
82:
83:     while ( !(option) )
84:     {
85:         pros::delay(50);
86:     }
87:
88:     if ( option == 1 )
89:     {
90:         lv_obj_set_hidden(warn_box, true);
91:         return true;
92:     }
93:
94:     else
95:     {
96:         lv_obj_set_hidden(warn_box, true);
97:         return false;
98:     }
99:
100: }
101:
102:
103:
104:
105:
106:
107: Loading::Loading()
108: {
109:     loader = lv_bar_create(lv_scr_act(), NULL);
110:     lv_obj_set_size(loader, 100, 20);
111:     lv_bar_set_value(loader, 1);
112: }
113:
```

```
114: Loading::~Loading()
115: {
116:     lv_obj_del(loader);
117: }
118:
119: /**
120:  * loader is shown on a parent at specified location
121:  * animation time is set by user, so this function only works if user knows
122:  * about how long the function will take
123:  * this function is meant as a filler so that if some initialization occurs
124:  * the gui does not appear like its hanging for no reason
125:  */
126: void Loading::show_load(int estimated_duration, lv_obj_t *parent, int x, int y)
127: {
128:     lv_obj_set_hidden(loader, false);
129:     lv_bar_set_value(loader, 1);
130:     lv_obj_set_parent(loader, parent);
131:     lv_obj_set_top(loader, true);
132:
133:     lv_obj_set_pos(loader, x, y);
134:
135:     lv_bar_set_value_anim(loader, 100, estimated_duration);
136: }
137:
138: /**
139:  * hides the loader for when the initialization by user is finished
140:  */
141: void Loading::hide_load()
142: {
143:     lv_obj_set_hidden(loader, true);
144: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/lcdCode/Styles.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains base class for styles of gui objects
8:   *
9:   */
10:
11: #ifndef __STYLES__
12: #define __STYLES__
13:
14:
15: #include "../../../include/main.h"
16:
17: //defines colors to use for each style
18: #define BLUE_BORDER LV_COLOR_BLUE
19: #define RED_BORDER LV_COLOR_RED
20: #define BG LV_COLOR_GRAY
21: #define BUTTON_REL LV_COLOR_SILVER
22: #define BUTTON_PR LV_COLOR_NAVY
23: #define TEXT LV_COLOR_WHITE
24: #define BODY_TEXT LV_COLOR_BLACK
25: #define SW_INDIC LV_COLOR_HEX(0x9fc8ef)
26:
27: //allows use of other fonts
28: #define USE_DEJAVU_12
29: #define USE_DEJAVU_16
30: #include "../../../include/fonts/fonts.h"
31:
32:
33: /**
34:  * @see: ../../include/fonts/fonts.hpp
35:  * @see ../fonts/
36:  *
37:  * base class that contains different colors and styles to be used throughout
38:  * the gui
39:  * designed so that there is no repetion of styles and so they are all in one place
40:  * designed to be inherited
41:  */
42: class Styles
43: {
44:     protected:
45:         //styles
46:         lv_style_t blue;
47:         lv_style_t red;
48:         lv_style_t gray;
49:
50:         lv_style_t toggle_btn_released;
51:         lv_style_t toggle_btn_pressed;
52:
53:         lv_style_t toggle_tabbtn_released;
54:         lv_style_t toggle_tabbtn_pressed;
55:
56:         lv_style_t sw_toggled;
57:         lv_style_t sw_off;
58:         lv_style_t sw_bg;
59:         lv_style_t sw_indic;
60:
61:         lv_style_t heading_text;
62:         lv_style_t body_text;
63:         lv_style_t subheading_text;
64:
65:         lv_style_t lines;
66:
67:         lv_style_t warn_box_bg;
68:         lv_style_t warn_box_pressed;
69:         lv_style_t warn_box_released;
70:
71:         lv_style_t loader_style;
72:
73:     public:
74:         Styles();
75:         virtual ~Styles();
76:
77: };
78:
79: #endif
```

```
 1: /**
 2:  * @file: ./RobotCode/src/objects/lcdCode/Styles.cpp
 3:  * @author: Aiden Carney
 4:  * @reviewed_on: 10/15/2019
 5:  * @reviewed_by: Aiden Carney
 6:  *
 7:  * @see: Styles.hpp
 8:  *
 9:  * contains base class for styles for gui
10:  *
11:  */
12:
13: #include "../../../include/main.h"
14: #include "../../../include/api.h"
15:
16: #include "Styles.hpp"
17:
18:
19: Styles::Styles()
20: {
21: //red style
22:     lv_style_copy(&red, &lv_style_scr);
23:     red.body.main_color = LV_COLOR_RED;
24:     red.body.grad_color = LV_COLOR_RED;
25:     red.body.border.color = LV_COLOR_RED;
26:
27: //blue style
28:     lv_style_copy(&blue, &lv_style_scr);
29:     blue.body.main_color = LV_COLOR_BLUE;
30:     blue.body.grad_color = LV_COLOR_BLUE;
31:     blue.body.border.color = LV_COLOR_BLUE;
32:
33: //gray style
34:     lv_style_copy(&gray, &lv_style_scr);
35:     gray.body.main_color = BG;
36:     gray.body.grad_color = BG;
37:     gray.body.border.color = BG;
38:     gray.body.border.width = 10;
39:
40: //style for when the button is not pressed
41:     lv_style_copy(&toggle_btn_released, &lv_style_plain);
42:     toggle_btn_released.body.main_color = BUTTON_REL;
43:     toggle_btn_released.body.grad_color = BUTTON_REL;
44:     toggle_btn_released.body.border.color = BUTTON_REL;
45:     toggle_btn_released.body.border.width = 2;
46:     toggle_btn_released.body.border.opa = LV_OPA_0;
47:     toggle_btn_released.body.radius = 5;
48:     toggle_btn_released.text.color = TEXT;
49:
50: //style for when the button is pressed
51:     lv_style_copy(&toggle_btn_pressed, &lv_style_plain);
52:     toggle_btn_pressed.body.main_color = BUTTON_PR;
53:     toggle_btn_pressed.body.grad_color = BUTTON_PR;
54:     toggle_btn_pressed.body.border.color = BUTTON_REL;
55:     toggle_btn_pressed.text.color = TEXT;
56:
57: //style for when tabview button is not pressed
58:     lv_style_copy(&toggle_tabbtn_released, &lv_style_plain);
59:     toggle_tabbtn_released.body.main_color = BUTTON_REL;
60:     toggle_tabbtn_released.body.grad_color = BUTTON_REL;
61:     toggle_tabbtn_released.body.border.color = BUTTON_REL;
62:     toggle_tabbtn_released.body.border.width = 2;
63:     toggle_tabbtn_released.body.border.opa = LV_OPA_0;
64:     toggle_tabbtn_released.text.color = TEXT;
65:     toggle_tabbtn_released.text.font = &dejavu_12;
66:
67: //style for when tabview button is pressed
68:     lv_style_copy(&toggle_tabbtn_pressed, &lv_style_plain);
69:     toggle_tabbtn_pressed.body.main_color = BUTTON_PR;
70:     toggle_tabbtn_pressed.body.grad_color = BUTTON_PR;
71:     toggle_tabbtn_pressed.body.border.color = BUTTON_REL;
72:     toggle_tabbtn_pressed.text.color = TEXT;
73:     toggle_tabbtn_pressed.text.font = &dejavu_12;
74:
75: //switch on
76:     lv_style_copy(&sw_toggled, &lv_style_pretty_color);
77:     sw_toggled.body.radius = LV_RADIUS_CIRCLE;
78:     sw_toggled.body.shadow.width = 4;
79:     sw_toggled.body.shadow.type = LV_SHADOW_BOTTOM;
80:
81: //switch off
82:     lv_style_copy(&sw_off, &lv_style_pretty);
83:     sw_off.body.radius = LV_RADIUS_CIRCLE;
84:     sw_off.body.shadow.width = 4;
85:     sw_off.body.shadow.type = LV_SHADOW_BOTTOM;
86:
87: //switch background
88:     lv_style_copy(&sw_bg, &lv_style_pretty);
89:     sw_bg.body.radius = LV_RADIUS_CIRCLE;
90:
91: //switch indicator
92:     lv_style_copy(&sw_indic, &lv_style_pretty_color);
93:     sw_indic.body.radius = LV_RADIUS_CIRCLE;
94:     sw_indic.body.main_color = SW_INDIC;
95:     sw_indic.body.grad_color = SW_INDIC;
96:     sw_indic.body.padding.hor = 0;
97:     sw_indic.body.padding.ver = 0;
98:
99: //heading text
100:     lv_style_copy(&heading_text, &lv_style_plain);
101:     heading_text.text.letter_space = 2;
102:     heading_text.text.line_space = 1;
103:     heading_text.text.color = TEXT;
104:     heading_text.text.font = &lv_font_dejavu_20;
105:
106: //body text
107:     lv_style_copy(&body_text, &lv_style_plain);
108:     body_text.text.letter_space = 2;
109:     body_text.text.line_space = 1;
110:     body_text.text.color = BODY_TEXT;
111:     body_text.text.font = &dejavu_12;
112:
113: //subheading text
```

```
114:     lv_style_copy(&subheading_text, &lv_style_plain);
115:     subheading_text.text.letter_space = 2;
116:     subheading_text.text.line_space = 1;
117:     subheading_text.text.color = BODY_TEXT;
118:     subheading_text.text.font = &dejavu_16;
119:
120:   //style for lines
121:     lv_style_copy(&lines, &lv_style_plain);
122:     lines.line.color = BUTTON_PR;
123:     lines.line.width = 5;
124:
125:   //styles for warning box
126:     //background
127:     lv_style_copy(&warn_box_bg, &lv_style_pretty);
128:     warn_box_bg.body.main_color = LV_COLOR_MAKE(0xf5, 0x45, 0x2e);
129:     warn_box_bg.body.grad_color = LV_COLOR_MAKE(0xb9, 0x1d, 0x09);
130:     warn_box_bg.body.border.color = LV_COLOR_MAKE(0x3f, 0x0a, 0x03);
131:     warn_box_bg.text.color = LV_COLOR_WHITE;
132:     warn_box_bg.body.padding.hor = 12;
133:     warn_box_bg.body.padding.ver = 8;
134:     warn_box_bg.body.shadow.width = 8;
135:
136:     //button not pressed
137:     lv_style_copy(&warn_box_released, &lv_style_btn_rel);
138:     warn_box_released.body.empty = 1;
139:     warn_box_released.body.border.color = LV_COLOR_WHITE;
140:     warn_box_released.body.border.width = 2;
141:     warn_box_released.body.border.opa = LV_OPA_70;
142:     warn_box_released.body.padding.hor = 12;
143:     warn_box_released.body.padding.ver = 8;
144:
145:     //button being pressed
146:     lv_style_copy(&warn_box_pressed, &warn_box_released);
147:     warn_box_pressed.body.empty = 0;
148:     warn_box_pressed.body.main_color = LV_COLOR_MAKE(0x5d, 0x0f, 0x04);
149:     warn_box_pressed.body.grad_color = LV_COLOR_MAKE(0x5d, 0x0f, 0x04);
150:
151:   //style for loader
152:     lv_style_copy(&loader_style, &lv_style_plain);
153:     loader_style.line.width = 10;  //10 px thick arc
154:     loader_style.line.color = LV_COLOR_HEX3(0x258); //Blueish arc color
155:
156:     loader_style.body.border.color = LV_COLOR_HEX3(0xBBB); //Gray background color
157:     loader_style.body.border.width = 10;
158:     loader_style.body.padding.hor = 0;
159:
160:   }
161:
162:   Styles::~Styles()
163:   {
164:
165:   }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/lcdCode/TemporaryScreen.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   * TODO: deprecate, possibly move somewhere else, file does very little and could be merged elsewhere
7:   *
8:   * contains a global static screen that can be loaded so that the one screen needs to
9:   * be loaded at all times rule is not broken
10:  *
11:  */
12:
13: #ifndef __TEMPORARYSCREEN_HPP__
14: #define __TEMPORARYSCREEN_HPP__
15:
16: #include "../../../include/main.h"
17:
18:
19: struct tempScreen
20: {
21:     static lv_obj_t *temp_screen;
22: };
23:
24:
25:
26: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/lcdCode/TemporaryScreen.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see: TemporaryScreen.hpp
8:   *
9:   * global screen part of a struct that can be loaded
10:  * has no parent so that it is always valid
11:  *
12:  */
13:
14: #include "TemporaryScreen.hpp"
15: #include "Styles.hpp"
16: #include "../../../include/main.h"
17:
18:
19: lv_obj_t *tempScreen::temp_screen = lv_obj_create(NULL, NULL);
```

```
 1: /**
 2:  * @file: ./RobotCode/src/objects/lcdCode/gui.hpp
 3:  * @author: Aiden Carney
 4:  * @reviewed_on: 10/15/2019
 5:  * @reviewed_by: Aiden Carney
 6:  * TODO: clean up conditionals, add config file
 7:  *
 8:  * contains auton selector gui selection all put together in one function
 9:  *
10:  */
11: #ifndef __GUI_HPP__
12: #define __GUI_HPP__
13:
14:
15: #include "../include/main.h"
16:
17: #include "AutonSelection/SelectionScreen.hpp"
18: #include "AutonSelection/OptionsScreen.hpp"
19: #include "AutonSelection/PrepScreen.hpp"
20: #include "AutonSelection/ActionsScreen.hpp"
21: #include "DriverControl/DriverControlLCD.hpp"
22: #include "../../DriverControl.hpp"
23: #include "Debug/Debug.hpp"
24: #include "TemporaryScreen.hpp"
25:
26:
27: /**
28:  * @return: int -> number of auton selected
29:  *
30:  * @see: ./AutonSelection
31:  * @see: ./Debug
32:  *
33:  * TODO: add more meaningful config options, clean up conditionals
34:  *
35:  * iterates and interacts with user to find final auton choice, and config options
36:  *
37:  */
38: int chooseAuton();
39:
40:
41:
42:
43:
44: #endif
```

```
  1: /**
  2:  * @file: ./RobotCode/src/objects/lcdCode/gui.cpp
  3:  * @author: Aiden Carney
  4:  * @reviewed_on: 10/25/2019
  5:  * @reviewed_by: Aiden Carney
  6:  *
  7:  * @see: gui.hpp
  8:  *
  9:  * contains implementation of gui
 10:  *
 11:  */
 12:
 13: #include "../../../include/main.h"
 14:
 15: #include "gui.hpp"
 16: #include "../../Autons.hpp"
 17: #include "../motors/MotorThread.hpp"
 18: #include "../../DriverControl.hpp"
 19: #include "TemporaryScreen.hpp"
 20:
 21:
 22: /**
 23:  * iterates through selecting for user to go through stages selecting an auton or the debugger
 24:  * and then all the config options
 25:  * loads all screens at start so there are no mem management issues
 26:  * finishes when all options are chosen
 27:  */
 28: int chooseAuton()
 29: {
 30:     Autons auton_data;
 31:
 32:     //init screens so that loading time is faster
 33:     SelectionScreen scr1;
 34:     OptionsScreen scr2;
 35:     PrepScreen scr3;
 36:
 37:     int finalAutonChoice = 0;
 38:     int auton = 1;
 39:     bool confirm = false;
 40:     int interval = 20;
 41:
 42:     while ( !(finalAutonChoice) ) //allows user to go to previous screen
 43:     {
 44:         scr2.back = false;
 45:
 46:         auton = scr1.selectAuton( auton ); //get auton option
 47:
 48:         if ( auton == auton_data.driver_control_num ) {  //if prog with no auton is selected
 49:             finalAutonChoice = 1;
 50:         } else if ( auton == auton_data.debug_auton_num ) {  //if debugger is selected
 51:             //starts driver control for debugging purposes
 52:             MotorThread* motor_thread = MotorThread::get_instance();
 53:             Motors::register_motors();
 54:             motor_thread->start_thread();
 55:
 56:             pros::Task driver_control_task (driver_control,
 57:                                             (void*)NULL,
 58:                                             TASK_PRIORITY_DEFAULT,
 59:                                             TASK_STACK_DEPTH_DEFAULT,
 60:                                             "DriverControlTask");
 61:
 62:             debug();
 63:
 64:             //ends driver control because it should not be enabled when
 65:             //auton is being selected
 66:             driver_control_task.remove();
 67:
 68:             Motors::unregister_motors();
 69:             motor_thread->stop_thread();
 70:         } else {
 71:             finalAutonChoice = auton;
 72:         }
 73:
 74:         // selection screen has been removed temporarily because options are not in use
 75:         // else
 76:         // {
 77:         //     while ( !(scr2.back) && !(finalAutonChoice) )
 78:         //     //if user selects a program with an auton
 79:         //     {
 80:         //         autonConfig cnfg = scr2.getOptions( auton ); //get config options
 81:         //
 82:         //         if ( !(scr2.back) ) //if user does not want to go back from screen 2
 83:         //         {
 84:         //
 85:         //             scr3.getConfirmation( auton ); //gets confirmation from user
 86:         //             if ( scr3.confirm )
 87:         //             {
 88:         //                 finalAutonChoice = auton;
 89:         //             }
 90:         //
 91:         //
 92:         //         }
 93:         //         else
 94:         //         {
 95:         //             break;
 96:         //         }
 97:         //
 98:         //
 99:         //     }
100:         // }
101:
102:     }
103:
104:
105:
106:     return finalAutonChoice;
107: }
```

```
 1: /**
 2:  * @file: ./RobotCode/src/lcdCode/DriverControl/AutonomousLCD.hpp
 3:  * @author: Aiden Carney
 4:  * @reviewed_on:
 5:  * @reviewed_by:
 6:  *
 7:  * contains the lcd screen used during auton
 8:  *
 9:  */
10: #ifndef __AUTONOMOUSLCD_HPP__
11: #define __AUTONOMOUSLCD_HPP__
12:
13: #include "main.h"
14:
15: #include "../Styles.hpp"
16:
17:
18: /**
19:  * @see: ../Styles.hpp
20:  *
21:  * contains lcd to be used during driver control
22:  */
23: class AutonomousLCD : private Styles
24: {
25:     private:
26:         lv_obj_t *screen;
27:
28:         lv_obj_t *logo_img;
29:
30:
31:         //labels
32:         lv_obj_t *title_label;
33:         lv_obj_t *auton_label;
34:         lv_obj_t *description_label;
35:
36:
37:     public:
38:         AutonomousLCD();
39:         ~AutonomousLCD();
40:
41:
42:         /**
43:          * @param: int auton_number -> the autonomous number
44:          * @return: None
45:          *
46:          * TODO: add actual content to be updated
47:          *
48:          * function to be used to update the gui to keep data relevant
49:          */
50:         void update_labels(int auton_number);
51:
52:
53:         void log_to_lcd(std::string msg);
54:
55:
56: };
57:
58: #endif
```

```
  1: /**
  2:  * @file: ./RobotCode/src/lcdCode/DriverControl/AutonomousLCD.cpp
  3:  * @author: Aiden Carney
  4:  * @reviewed_on:
  5:  * @reviewed_by:
  6:  *
  7:  * @see: DriverControlLCD.hpp
  8:  *
  9:  * contains methods for driver control lcd
 10:  */
 11:
 12: #include "main.h"
 13:
 14: #include "../../../Autons.hpp"
 15: #include "../../serial/Logger.hpp"
 16: #include "../../position_tracking/PositionTracker.hpp"
 17: #include "../AutonSelection/OptionsScreen.hpp"
 18: #include "../Debug/Debug.hpp"
 19: #include "AutonomousLCD.hpp"
 20:
 21: LV_IMG_DECLARE(logo);
 22:
 23:
 24: AutonomousLCD::AutonomousLCD()
 25: {
 26:     screen = lv_obj_create(NULL, NULL);
 27:     lv_obj_set_style(screen, &gray);
 28:
 29: // init labels
 30:     title_label = lv_label_create(screen, NULL);
 31:     lv_obj_set_style(title_label, &heading_text);
 32:     lv_obj_set_width(title_label, 300);
 33:     lv_obj_set_height(title_label, 20);
 34:     lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
 35:     lv_label_set_text(title_label, "Autonomous");
 36:
 37:
 38:     auton_label = lv_label_create(screen, NULL);
 39:     lv_obj_set_style(auton_label, &subheading_text);
 40:     lv_obj_set_width(auton_label, 300);
 41:     lv_obj_set_height(auton_label, 20);
 42:     lv_label_set_align(auton_label, LV_LABEL_ALIGN_CENTER);
 43:     lv_label_set_text(auton_label, "");
 44:
 45:
 46:     description_label = lv_label_create(screen, NULL);
 47:     lv_obj_set_style(description_label, &subheading_text);
 48:     lv_obj_set_width(description_label, 300);
 49:     lv_obj_set_height(description_label, 20);
 50:     lv_label_set_align(description_label, LV_LABEL_ALIGN_LEFT);
 51:     lv_label_set_text(description_label, "");
 52:
 53:
 54: // init image area
 55:     logo_img = lv_img_create(screen, NULL);
 56:     lv_img_set_src(logo_img, &logo);
 57:     lv_img_set_auto_size(logo_img, false);
 58:     lv_obj_set_width(logo_img, 210);
 59:     lv_obj_set_height(logo_img, 150);
 60:
 61:
 62: // place objects
 63:     lv_obj_set_pos(title_label, 180, 9);
 64:
 65:     lv_obj_set_pos(logo_img, 30, 40);
 66:
 67:     lv_obj_set_pos(auton_label, 280, 40);
 68:     lv_obj_set_pos(description_label, 280, 80);
 69:
 70: }
 71:
 72:
 73:
 74: AutonomousLCD::~AutonomousLCD()
 75: {
 76:     lv_obj_del(screen);
 77: }
 78:
 79:
 80:
 81:
 82: /**
 83:  * updates colors and borders during driver control
 84:  * keeps data relevent
 85:  */
 86: void AutonomousLCD::update_labels(int auton_number)
 87: {
 88:     Autons autons;
 89:
 90:     lv_scr_load(screen);
 91:
 92:     lv_label_set_text(auton_label, autons.AUTONOMOUS_NAMES.at(auton_number));
 93:     lv_label_set_text(description_label, autons.AUTONOMOUS_DESCRIPTIONS.at(auton_number));
 94: }
 95:
 96:
 97: void AutonomousLCD::log_to_lcd(std::string msg) {
 98:     lv_scr_load(screen);
 99:     lv_label_set_text(description_label, msg.c_str());
100: }
```

```
  1: /**
  2:  * @file: ./RobotCode/src/lcdCode/DriverControl/DriverControlLCD.hpp
  3:  * @author: Aiden Carney
  4:  * @reviewed_on: 10/15/2019
  5:  * @reviewed_by: Aiden Carney
  6:  * TODO: add actual content instead of blank screen
  7:  *
  8:  * contains the lcd screen used during driver control
  9:  *
 10:  */
 11: #ifndef __DRIVERCONTROLLCD_HPP__
 12: #define __DRIVERCONTROLLCD_HPP__
 13:
 14: #include "main.h"
 15:
 16: #include "../Styles.hpp"
 17:
 18:
 19: /**
 20:  * @see: ../Styles.hpp
 21:  *
 22:  * contains lcd to be used during driver control
 23:  */
 24: class DriverControlLCD : private Styles
 25: {
 26:     private:
 27:         static bool log_data;
 28:         static bool open_debugger;
 29:         static std::string toggle_logging_text;
 30:
 31:         lv_obj_t *screen;
 32:
 33:         lv_obj_t *logo_img;
 34:
 35:
 36:         //labels
 37:         lv_obj_t *title_label;
 38:         lv_obj_t *queue_size_label;
 39:
 40:         //buttons
 41:         lv_obj_t *btn_debugger;
 42:         lv_obj_t *btn_run_auton;
 43:         lv_obj_t *btn_toggle_logging;
 44:         lv_obj_t *btn_flush_queue;
 45:
 46:         lv_obj_t *btn_debugger_label;
 47:         lv_obj_t *btn_run_auton_label;
 48:         lv_obj_t *btn_toggle_logging_label;
 49:         lv_obj_t *btn_flush_queue_label;
 50:
 51:
 52:         /**
 53:          * @param: lv_obj_t* btn -> button that called the funtion
 54:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
 55:          *
 56:          * button callback function used to open debugger
 57:          */
 58:         static lv_res_t btn_debugger_action(lv_obj_t *btn);
 59:
 60:
 61:         /**
 62:          * @param: lv_obj_t* btn -> button that called the funtion
 63:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
 64:          *
 65:          * button callback function used to run auton
 66:          */
 67:         static lv_res_t btn_run_auton_action(lv_obj_t *btn);
 68:
 69:
 70:         /**
 71:          * @param: lv_obj_t* btn -> button that called the funtion
 72:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
 73:          *
 74:          * button callback function used to toggle logging of motors and other various items
 75:          */
 76:         static lv_res_t btn_toggle_logging_action(lv_obj_t *btn);
 77:
 78:
 79:         /**
 80:          * @param: lv_obj_t* btn -> button that called the funtion
 81:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
 82:          *
 83:          * button callback function used to flush the logging queue
 84:          */
 85:         static lv_res_t btn_flush_queue_action(lv_obj_t *btn);
 86:
 87:
 88:     public:
 89:         DriverControlLCD();
 90:         ~DriverControlLCD();
 91:
 92:
 93:         /**
 94:          * @return: None
 95:          *
 96:          * TODO: add actual content to be updated
 97:          *
 98:          * function to be used to update the gui to keep data relevant
 99:          */
100:         void update_labels();
101:
102:
103: };
104:
105: #endif
```

```
/**
 * @file: ./RobotCode/src/lcdCode/DriverControl/DriverControlLCD.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: DriverControlLCD.hpp
 *
 * contains methods for driver control lcd
 */

#include "main.h"

#include "../../../Autons.hpp"
#include "../../serial/Logger.hpp"
#include "../../position_tracking/PositionTracker.hpp"
#include "../AutonSelection/OptionsScreen.hpp"
#include "../Debug/Debug.hpp"
#include "DriverControlLCD.hpp"


bool DriverControlLCD::log_data = false;
bool DriverControlLCD::open_debugger = false;
std::string DriverControlLCD::toggle_logging_text = "Start Logging";
LV_IMG_DECLARE(logo);


DriverControlLCD::DriverControlLCD()
{
    log_data = false;

    screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(screen, &gray);

// init labels
    queue_size_label = lv_label_create(screen, NULL);
    lv_obj_set_style(queue_size_label, &subheading_text);
    lv_obj_set_width(queue_size_label, 300);
    lv_obj_set_height(queue_size_label, 20);
    lv_label_set_align(queue_size_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(queue_size_label, "Logger Queue Size: ");


    title_label = lv_label_create(screen, NULL);
    lv_obj_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 300);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Driver Control");

// init image area
    logo_img = lv_img_create(screen, NULL);
    lv_img_set_src(logo_img, &logo);
    lv_img_set_auto_size(logo_img, false);
    lv_obj_set_width(logo_img, 210);
    lv_obj_set_height(logo_img, 150);

// init buttons
    //button
    btn_debugger = lv_btn_create(screen, NULL);
    lv_btn_set_style(btn_debugger, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_debugger, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_debugger, LV_BTN_ACTION_CLICK, btn_debugger_action);
    lv_obj_set_width(btn_debugger, 180);
    lv_obj_set_height(btn_debugger, 25);

    //label
    btn_debugger_label = lv_label_create(btn_debugger, NULL);
    lv_obj_set_style(btn_debugger_label, &subheading_text);
    lv_label_set_text(btn_debugger_label, "Open Debugger");


    //button
    btn_run_auton = lv_btn_create(screen, NULL);
    lv_btn_set_style(btn_run_auton, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_run_auton, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_run_auton, LV_BTN_ACTION_CLICK, btn_run_auton_action);
    lv_obj_set_width(btn_run_auton, 180);
    lv_obj_set_height(btn_run_auton, 25);

    //label
    btn_run_auton_label = lv_label_create(btn_run_auton, NULL);
    lv_obj_set_style(btn_run_auton_label, &subheading_text);
    lv_label_set_text(btn_run_auton_label, "Run Auton");


    //button
    btn_toggle_logging = lv_btn_create(screen, NULL);
    lv_btn_set_style(btn_toggle_logging, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_toggle_logging, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_toggle_logging, LV_BTN_ACTION_CLICK, btn_toggle_logging_action);
    lv_obj_set_width(btn_toggle_logging, 180);
    lv_obj_set_height(btn_toggle_logging, 25);

    //label
    btn_toggle_logging_label = lv_label_create(btn_toggle_logging, NULL);
    lv_obj_set_style(btn_toggle_logging_label, &subheading_text);
    lv_label_set_text(btn_toggle_logging_label, toggle_logging_text.c_str());


    //button
    btn_flush_queue = lv_btn_create(screen, NULL);
    lv_btn_set_style(btn_flush_queue, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_flush_queue, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_flush_queue, LV_BTN_ACTION_CLICK, btn_flush_queue_action);
    lv_obj_set_width(btn_flush_queue, 180);
    lv_obj_set_height(btn_flush_queue, 25);

    //label
    btn_flush_queue_label = lv_label_create(btn_flush_queue, NULL);
    lv_obj_set_style(btn_flush_queue_label, &subheading_text);
    lv_label_set_text(btn_flush_queue_label, "Flush Logger Queue");

```

```
114:
115:  // place objects
116:     lv_obj_set_pos(title_label, 180, 9);
117:     lv_obj_set_pos(queue_size_label, 280, 200);
118:     lv_obj_set_pos(logo_img, 30, 40);
119:
120:     lv_obj_set_pos(btn_debugger, 280, 40);
121:     lv_obj_set_pos(btn_run_auton, 280, 80);
122:     lv_obj_set_pos(btn_toggle_logging, 280, 120);
123:     lv_obj_set_pos(btn_flush_queue, 280, 160);
124:
125:  }
126:
127:
128:
129:  DriverControlLCD::~DriverControlLCD()
130:  {
131:     lv_obj_del(screen);
132:  }
133:
134:
135:
136:
137:  lv_res_t DriverControlLCD::btn_debugger_action(lv_obj_t *btn)
138:  {
139:     open_debugger = true;
140:     return LV_RES_OK;
141:  }
142:
143:
144:
145:  lv_res_t DriverControlLCD::btn_run_auton_action(lv_obj_t *btn)
146:  {
147:     pros::delay(3000);
148:
149:     Autons auton_obj;
150:
151:     Motors::disable_driver_control();
152:
153:     auton_obj.run_autonomous();
154:
155:     Motors::enable_driver_control();
156:
157:     return LV_RES_OK;
158:  }
159:
160:
161:
162:  lv_res_t DriverControlLCD::btn_toggle_logging_action(lv_obj_t *btn)
163:  {
164:     if(log_data)
165:     {
166:        log_data = false;
167:     }
168:     else
169:     {
170:        log_data = true;
171:     }
172:     return LV_RES_OK;
173:  }
174:
175:
176:
177:  lv_res_t DriverControlLCD::btn_flush_queue_action(lv_obj_t *btn)
178:  {
179:     Logger logger;
180:     while(logger.get_count() > 0)
181:     {
182:        logger.dump();
183:     }
184:     return LV_RES_OK;
185:  }
186:
187:
188:
189:
190:
191:  /**
192:   * updates colors and borders during driver control
193:   * keeps data relevent
194:   */
195:  void DriverControlLCD::update_labels()
196:  {
197:     lv_scr_load(screen);
198:
199:     Logger logger;
200:     // PositionTracker* pos_tracker = PositionTracker::get_instance();
201:
202:     // update toggle logging label text
203:     if(log_data)
204:     {
205:        lv_label_set_text(btn_toggle_logging_label, "Stop Logging");
206:        Motors::set_log_level(1);
207:
208:        // pos_tracker->start_logging();
209:
210:        Sensors::log_data();
211:     }
212:     else
213:     {
214:        lv_label_set_text(btn_toggle_logging_label, "Start Logging");
215:        Motors::set_log_level(0);
216:
217:        // pos_tracker->stop_logging();
218:     }
219:
220:     if(open_debugger)
221:     {
222:        debug();
223:        open_debugger = false;
224:     }
225:
226:     //update logger queue size label
```

```
227:     std::string text = std::string("Logger Queue Size: ") + std::to_string(logger.get_count());
228:     lv_label_set_text(queue_size_label, text.c_str());
229: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/AutonSelecton/ActionsScreen.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   * TODO: add actual content for when actions are decided on
7:   *
8:   * does nothing
9:   *
10:  */
11:
12: #ifndef __ACTIONSSCREEN_HPP__
13: #define __ACTIONSSCREEN_HPP__
14:
15:
16: #include "../include/main.h"
17:
18:
19:
20:
21: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/AutonSelecton/ActionsScreen.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see: ActionsScreen.hpp
8:   *
9:   * does nothing
10:  *
11:  */
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/AutonSelecton/OptionsScreen.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   * TODO: add correct options when they are decided on, deprecate static options to reduce coupling
7:   *
8:   * contains class with methods to decide on options for auton
9:   *
10:  */
11:
12:  #ifndef __OPTIONSSCREEN_HPP__
13:  #define __OPTIONSSCREEN_HPP__
14:
15:
16:  #include "../../../../include/main.h"
17:
18:  #include "../../../Autons.hpp"
19:  #include "../Styles.hpp"
20:
21:  /**
22:   * @see: ../Styles.hpp
23:   * @see: ./AutonSelection
24:   * @see: ../gui.hpp
25:   *
26:   * contains methods to get options for auton period
27:   */
28:  class OptionsScreen : private Styles
29:  {
30:    private:
31:      //screen
32:      lv_obj_t *options_screen;
33:
34:      //labels
35:      lv_obj_t *title_label;
36:
37:      lv_obj_t *sw_use_hardcoded_label;
38:      lv_obj_t *sw_gyro_turn_label;
39:      lv_obj_t *sw_accelleration_ctrl_label;
40:      lv_obj_t *sw_check_motor_tmp_label;
41:      lv_obj_t *sw_use_previous_macros_label;
42:      lv_obj_t *sw_record_label;
43:
44:      //buttons
45:      lv_obj_t *btn_confirm;
46:      lv_obj_t *btn_back;
47:
48:      //button labels
49:      lv_obj_t *btn_confirm_label;
50:      lv_obj_t *btn_back_label;
51:
52:
53:      //switches
54:      lv_obj_t *sw_use_hardcoded;
55:      lv_obj_t *sw_gyro_turn;
56:      lv_obj_t *sw_accelleration_ctrl;
57:      lv_obj_t *sw_check_motor_tmp;
58:      lv_obj_t *sw_use_previous_macros;
59:      lv_obj_t *sw_record;
60:
61:
62:    public:
63:      OptionsScreen();
64:      ~OptionsScreen();
65:
66:      static autonConfig cnfg;
67:      static bool nextScreen;
68:      static bool back;
69:
70:
71:      //button functions
72:
73:      /**
74:       * @param: lv_obj_t* btn -> button that called the funtion
75:       * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
76:       *
77:       * button callback function used to set variable so that gui continues
78:       * to the next stage
79:       */
80:      static lv_res_t btn_confirm_action(lv_obj_t *btn);
81:
82:      /**
83:       * @param: lv_obj_t* btn -> button that called the funtion
84:       * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
85:       *
86:       * button callback function used to set variable so that gui goes back to
87:       * the previous stage
88:       */
89:      static lv_res_t btn_back_action(lv_obj_t *btn);
90:
91:
92:      //switch functions
93:
94:      /**
95:       * @param: lv_obj_t* sw -> switch object that was selected
96:       * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
97:       * TODO: merge with other functions to condense code and make it more expandable
98:       *
99:       * sets configuration option for using a compiled auton vs auton written on sd card
100:      */
101:     static lv_res_t sw_use_hardcoded_action(lv_obj_t *sw);
102:
103:     /**
104:      * @param: lv_obj_t* sw -> switch object that was selected
105:      * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
106:      * TODO: merge with other functions to condense code and make it more expandable
107:      *
108:      * sets configuration option for using gyro turns in auton
109:      */
110:     static lv_res_t sw_gyro_turn_action(lv_obj_t *sw);
111:
112:     /**
113:      * @param: lv_obj_t* sw -> switch object that was selected
```

```
114:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
115:         * TODO: merge with other functions to condense code and make it more expandable
116:         *
117:         * sets configuration option for using acceleration control code
118:         */
119:        static lv_res_t sw_accelleration_ctrl_action(lv_obj_t *sw);
120:
121:        /**
122:         * @param: lv_obj_t* sw -> switch object that was selected
123:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
124:         * TODO: merge with other functions to condense code and make it more expandable
125:         *
126:         * sets configuration option for limiting motor output based on temperature during the match
127:         */
128:        static lv_res_t sw_check_motor_tmp_action(lv_obj_t *sw);
129:
130:        /**
131:         * @param: lv_obj_t* sw -> switch object that was selected
132:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
133:         * TODO: merge with other functions to condense code and make it more expandable
134:         *
135:         * sets configuration option for allowing the use of previously recorded macros
136:         */
137:        static lv_res_t sw_use_previous_macros_action(lv_obj_t *sw);
138:
139:        /**
140:         * @param: lv_obj_t* sw -> switch object that was selected
141:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
142:         * TODO: merge with other functions to condense code and make it more expandable
143:         *
144:         * sets configuration option for recording match in the macro format
145:         */
146:        static lv_res_t sw_record_action(lv_obj_t *sw);
147:
148:
149:
150:        /**
151:         * @param: int auton -> number of auton selected, used to set color background of lcd
152:         * @return: autonConfig -> configuration struct with options based on how the switches were set
153:         *
154:         * @see: ../Structs.hpp
155:         * @see: ../Gui.hpp
156:         *
157:         * allows user to interact with the switches to set configuration options
158:         * user can choose to go back or continue with the options selected
159:         */
160:        autonConfig getOptions( int auton );
161:
162:    };
163:
164:
165:    #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/AutonSelecton/OptionsScreen.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see: OptionsScreen.hpp
8:   *
9:   * contains class with methods for getting a configuration struct for how
10:  * auton will be run
11:  */
12:
13:  #include <unordered_map>
14:
15:  #include "../../../../include/main.h"
16:  #include "../../../../include/api.h"
17:
18:  #include "../../../Autons.hpp"
19:  #include "../../controller/controller.hpp"
20:  #include "OptionsScreen.hpp"
21:
22:
23:  autonConfig OptionsScreen::cnfg;
24:  bool OptionsScreen::nextScreen = false;
25:  bool OptionsScreen::back = false;
26:
27:
28:
29:  OptionsScreen::OptionsScreen()
30:  {
31:      nextScreen = false;
32:      back = false;
33:
34:      options_screen = lv_obj_create(NULL, NULL);
35:
36:  //use hard coded autonomous
37:      //switch
38:      sw_use_hardcoded = lv_sw_create(options_screen, NULL); //switch template
39:      lv_sw_set_style(sw_use_hardcoded, LV_SW_STYLE_BG, &sw_bg);
40:      lv_sw_set_style(sw_use_hardcoded, LV_SW_STYLE_INDIC, &sw_indic);
41:      lv_sw_set_style(sw_use_hardcoded, LV_SW_STYLE_KNOB_ON, &sw_toggled);
42:      lv_sw_set_style(sw_use_hardcoded, LV_SW_STYLE_KNOB_OFF, &sw_off);
43:
44:      lv_sw_set_action(sw_use_hardcoded, sw_use_hardcoded_action); //map action
45:      lv_obj_set_width(sw_use_hardcoded, 40); //witdth
46:      lv_obj_set_height(sw_use_hardcoded, 20); //height
47:
48:      //label
49:      sw_use_hardcoded_label = lv_label_create(options_screen, NULL);
50:      lv_label_set_style(sw_use_hardcoded_label, &heading_text);
51:      lv_obj_set_width(sw_use_hardcoded_label, 10);
52:      lv_obj_set_height(sw_use_hardcoded_label, 20);
53:      lv_label_set_align(sw_use_hardcoded_label, LV_LABEL_ALIGN_LEFT);
54:      lv_label_set_text(sw_use_hardcoded_label, "Use Hardcoded Auton");
55:
56:
57:
58:  //gyro turns
59:      //switch
60:      sw_gyro_turn = lv_sw_create(options_screen, sw_use_hardcoded);
61:      lv_sw_set_action(sw_gyro_turn, sw_gyro_turn_action);
62:      lv_obj_set_width(sw_gyro_turn, 40);
63:      lv_obj_set_height(sw_gyro_turn, 20);
64:
65:      //label
66:      sw_gyro_turn_label = lv_label_create(options_screen, NULL);
67:      lv_label_set_style(sw_gyro_turn_label, &heading_text);
68:      lv_obj_set_width(sw_gyro_turn_label, 300);
69:      lv_obj_set_height(sw_gyro_turn_label, 20);
70:      lv_label_set_align(sw_gyro_turn_label, LV_LABEL_ALIGN_LEFT);
71:      lv_label_set_text(sw_gyro_turn_label, "Use Gyro Turns");
72:
73:
74:  //acceleration control
75:      //switch
76:      sw_accelleration_ctrl = lv_sw_create(options_screen, sw_use_hardcoded);
77:      lv_sw_set_action(sw_accelleration_ctrl, sw_accelleration_ctrl_action);
78:      lv_obj_set_width(sw_accelleration_ctrl, 40);
79:      lv_obj_set_height(sw_accelleration_ctrl, 20);
80:
81:      //label
82:      sw_accelleration_ctrl_label = lv_label_create(options_screen, NULL);
83:      lv_label_set_style(sw_accelleration_ctrl_label, &heading_text);
84:      lv_obj_set_width(sw_accelleration_ctrl_label, 300);
85:      lv_obj_set_height(sw_accelleration_ctrl_label, 20);
86:      lv_label_set_align(sw_accelleration_ctrl_label, LV_LABEL_ALIGN_LEFT);
87:      lv_label_set_text(sw_accelleration_ctrl_label, "Use Acceleration Control");
88:
89:
90:  //check motor temp
91:      //switch
92:      sw_check_motor_tmp = lv_sw_create(options_screen, sw_use_hardcoded);
93:      lv_sw_set_action(sw_check_motor_tmp, sw_check_motor_tmp_action);
94:      lv_obj_set_width(sw_check_motor_tmp, 40);
95:      lv_obj_set_height(sw_check_motor_tmp, 20);
96:
97:      //label
98:      sw_check_motor_tmp_label = lv_label_create(options_screen, NULL);
99:      lv_label_set_style(sw_check_motor_tmp_label, &heading_text);
100:     lv_obj_set_width(sw_check_motor_tmp_label, 300);
101:     lv_obj_set_height(sw_check_motor_tmp_label, 20);
102:     lv_label_set_align(sw_check_motor_tmp_label, LV_LABEL_ALIGN_LEFT);
103:     lv_label_set_text(sw_check_motor_tmp_label, "Limit Motor Temp");
104:
105:
106: //use previous macros
107:     //switch
108:     sw_use_previous_macros = lv_sw_create(options_screen, sw_use_hardcoded);
109:     lv_sw_set_action(sw_use_previous_macros, sw_use_previous_macros_action);
110:     lv_obj_set_width(sw_use_previous_macros, 40);
111:     lv_obj_set_height(sw_use_previous_macros, 20);
112:
113:     //label
```

```
114:     sw_use_previous_macros_label = lv_label_create(options_screen, NULL);
115:     lv_label_set_style(sw_use_previous_macros_label, &heading_text);
116:     lv_obj_set_width(sw_use_previous_macros_label, 300);
117:     lv_obj_set_height(sw_use_previous_macros_label, 20);
118:     lv_label_set_align(sw_use_previous_macros_label, LV_LABEL_ALIGN_LEFT);
119:     lv_label_set_text(sw_use_previous_macros_label, "Use Previously Recorded Macros");
120:
121:
122:   //record
123:     //switch
124:     sw_record = lv_sw_create(options_screen, sw_use_hardcoded);
125:     lv_sw_set_action(sw_record, sw_record_action);
126:     lv_obj_set_width(sw_record, 40);
127:     lv_obj_set_height(sw_record, 20);
128:
129:     //label
130:     sw_record_label = lv_label_create(options_screen, NULL);
131:     lv_label_set_style(sw_record_label, &heading_text);
132:     lv_obj_set_width(sw_record_label, 300);
133:     lv_obj_set_height(sw_record_label, 20);
134:     lv_label_set_align(sw_record_label, LV_LABEL_ALIGN_LEFT);
135:     lv_label_set_text(sw_record_label, "Record Motor Movements");
136:
137:
138:   //confirm button
139:     //button
140:     btn_confirm = lv_btn_create(options_screen, NULL);
141:     lv_btn_set_style(btn_confirm, LV_BTN_STYLE_REL, &toggle_btn_released);
142:     lv_btn_set_style(btn_confirm, LV_BTN_STYLE_PR, &toggle_btn_pressed);
143:     lv_btn_set_action(btn_confirm, LV_BTN_ACTION_CLICK, btn_confirm_action);
144:     lv_obj_set_width(btn_confirm, 300);
145:     lv_obj_set_height(btn_confirm, 25);
146:
147:     //label
148:     btn_confirm_label = lv_label_create(btn_confirm, NULL);
149:     lv_obj_set_style(btn_confirm_label, &heading_text);
150:     lv_label_set_text(btn_confirm_label, "Confirm");
151:
152:   //back button
153:     //button
154:     btn_back = lv_btn_create(options_screen, NULL);
155:     lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
156:     lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
157:     lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
158:     lv_obj_set_width(btn_back, 50);
159:     lv_obj_set_height(btn_back, 25);
160:
161:     //label
162:     btn_back_label = lv_label_create(btn_back, NULL);
163:     lv_obj_set_style(btn_back_label, &heading_text);
164:     lv_label_set_text(btn_back_label, "Back");
165:
166:
167:   //title label
168:     title_label = lv_label_create(options_screen, NULL);
169:     lv_obj_set_style(title_label, &heading_text);
170:     lv_obj_set_width(title_label, 300);
171:     lv_obj_set_height(title_label, 20);
172:     lv_label_set_align(title_label, LV_LABEL_ALIGN_LEFT);
173:     lv_label_set_text(title_label, "Auton");
174:
175:   //set postition of widgets
176:     lv_obj_set_pos(sw_use_hardcoded, 400, 40);
177:     lv_obj_set_pos(sw_gyro_turn, 400, 65);
178:     lv_obj_set_pos(sw_accelleration_ctrl, 400, 90);
179:     lv_obj_set_pos(sw_check_motor_tmp, 400, 115);
180:     lv_obj_set_pos(sw_use_previous_macros, 400, 140);
181:     lv_obj_set_pos(sw_record, 400, 165);
182:
183:     lv_obj_set_pos(sw_use_hardcoded_label, 20, 40);
184:     lv_obj_set_pos(sw_gyro_turn_label, 20, 65);
185:     lv_obj_set_pos(sw_accelleration_ctrl_label, 20, 90);
186:     lv_obj_set_pos(sw_check_motor_tmp_label, 20, 115);
187:     lv_obj_set_pos(sw_use_previous_macros_label, 20, 140);
188:     lv_obj_set_pos(sw_record_label, 20, 165);
189:
190:     lv_obj_set_pos(btn_back, 40, 200);
191:     lv_obj_set_pos(btn_confirm, 100, 200);
192:     lv_obj_set_pos(title_label, 210, 20);
193:   }
194:
195:
196:   OptionsScreen::~OptionsScreen()
197:   {
198:     lv_obj_del(sw_use_hardcoded_label);
199:     lv_obj_del(sw_gyro_turn_label);
200:     lv_obj_del(sw_accelleration_ctrl_label);
201:     lv_obj_del(sw_check_motor_tmp_label);
202:     lv_obj_del(sw_use_previous_macros_label);
203:     lv_obj_del(sw_record_label);
204:
205:     lv_obj_del(sw_use_hardcoded);
206:     lv_obj_del(sw_gyro_turn);
207:     lv_obj_del(sw_accelleration_ctrl);
208:     lv_obj_del(sw_check_motor_tmp);
209:     lv_obj_del(sw_use_previous_macros);
210:     lv_obj_del(sw_record);
211:
212:     lv_obj_del(btn_back_label);
213:     lv_obj_del(btn_confirm_label);
214:     lv_obj_del(title_label);
215:
216:     lv_obj_del(btn_back);
217:     lv_obj_del(btn_confirm);
218:
219:     lv_obj_del(options_screen);
220:
221:   }
222:
223:
224:
225:   /**
226:    * sets nextScreen so that main loop will break and go to next stage
```

```
227:  */
228:  lv_res_t OptionsScreen::btn_confirm_action(lv_obj_t *btn)
229:  {
230:     nextScreen = true;
231:     back = false;
232:
233:     return LV_RES_OK;
234:  }
235:
236:  /**
237:   * sets nextScreen so that main loop will break and go to the previous stage
238:   */
239:  lv_res_t OptionsScreen::btn_back_action(lv_obj_t *btn)
240:  {
241:     nextScreen = true;
242:     back = true;
243:
244:     return LV_RES_OK;
245:  }
246:
247:
248:
249:
250:  /**
251:   * sets or clears config option for using hard coded autons based on the
252:   * switches previous state
253:   */
254:  lv_res_t OptionsScreen::sw_use_hardcoded_action(lv_obj_t *sw)
255:  {
256:     cnfg.use_hardcoded = !(cnfg.use_hardcoded);
257:     return LV_RES_OK;
258:  }
259:
260:  /**
261:   * sets or clears config option for using gyro turns based on the
262:   * switches previous state
263:   */
264:  lv_res_t OptionsScreen::sw_gyro_turn_action(lv_obj_t *sw)
265:  {
266:     cnfg.gyro_turn = !(cnfg.gyro_turn);
267:     return LV_RES_OK;
268:  }
269:
270:
271:  /**
272:   * sets or clears config option for using acceleration control code based on the
273:   * switches previous state
274:   */
275:  lv_res_t OptionsScreen::sw_accelleration_ctrl_action(lv_obj_t *sw)
276:  {
277:     cnfg.accelleration_ctrl = !(cnfg.accelleration_ctrl);
278:     return LV_RES_OK;
279:  }
280:
281:
282:  /**
283:   * sets or clears config option for limiting motor output based on the
284:   * switches previous state
285:   */
286:  lv_res_t OptionsScreen::sw_check_motor_tmp_action(lv_obj_t *sw)
287:  {
288:     cnfg.check_motor_tmp = !(cnfg.check_motor_tmp);
289:     return LV_RES_OK;
290:  }
291:
292:
293:  /**
294:   * sets or clears config option for allowing use of previously recorded macros based on the
295:   * switches previous state
296:   */
297:  lv_res_t OptionsScreen::sw_use_previous_macros_action(lv_obj_t *sw)
298:  {
299:     cnfg.use_previous_macros = !(cnfg.use_previous_macros);
300:     return LV_RES_OK;
301:  }
302:
303:
304:  /**
305:   * sets or clears config option for recorded the match as a macro based on the
306:   * switches previous state
307:   */
308:  lv_res_t OptionsScreen::sw_record_action(lv_obj_t *sw)
309:  {
310:     cnfg.record = !(cnfg.record);
311:     return LV_RES_OK;
312:  }
313:
314:
315:
316:  /**
317:   * runs loop where user can set auton configuration options with digital switches
318:   * loop breaks when user clicks the back or continue button
319:   * if user click the back button then the back flag is set
320:   */
321:  autonConfig OptionsScreen::getOptions( int auton )
322:  {
323:     Controller controllers;
324:     Autons auton_data;
325:
326:     lv_sw_off(sw_use_hardcoded); //reset switches and values
327:     lv_sw_on(sw_gyro_turn);
328:     lv_sw_on(sw_accelleration_ctrl);
329:     lv_sw_off(sw_check_motor_tmp);
330:     lv_sw_on(sw_use_previous_macros);
331:     lv_sw_off(sw_record);
332:
333:     cnfg.use_hardcoded = 0;
334:     cnfg.gyro_turn = 1;
335:     cnfg.accelleration_ctrl = 1;
336:     cnfg.check_motor_tmp = 0;
337:     cnfg.use_previous_macros = 1;
338:     cnfg.record = 0;
339:
```

```
340:
341:    lv_label_set_text(title_label, auton_data.AUTONOMOUS_NAMES.at(auton));
342:
343:    //load screen
344:    lv_scr_load(options_screen);
345:
346:    //set color of border
347:    std::string color = auton_data.AUTONOMOUS_COLORS.at(auton);
348:    if (color == "blue")
349:    {
350:       gray.body.border.color = BLUE_BORDER;
351:    }
352:    else if (color == "red")
353:    {
354:       gray.body.border.color = RED_BORDER;
355:    }
356:    else
357:    {
358:       gray.body.border.color = BG;
359:    }
360:
361:    lv_obj_set_style(options_screen, &gray);
362:
363:
364:    back = false;
365:    nextScreen = false;
366:
367:    pros::delay( 100 );  //add delay so that previous button clicks do not register
368:
369:    while ( !(nextScreen) )
370:    {
371:       //allow controller to press the buttons as well
372:       if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_A) )
373:       {
374:          btn_confirm_action( NULL );
375:          pros::delay(100);
376:       }
377:       else if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_B) )
378:       {
379:          btn_back_action( NULL );
380:          pros::delay(100);
381:       }
382:
383:       pros::delay(20);
384:    }
385:
386:    return cnfg;
387:
388: }
```

```
 1: /**
 2:  * @file: ./RobotCode/src/lcdCode/AutonSelecton/PrepScreen.hpp
 3:  * @author: Aiden Carney
 4:  * @reviewed_on: 10/15/2019
 5:  * @reviewed_by: Aiden Carney
 6:  * TODO: add actual preparation steps text
 7:  * TODO: decouple initilization string, make it more configurable
 8:  *
 9:  * contains class with methods for showing the things that will occur
10:  * before the auton is run
11:  */
12: 
13: #ifndef __PREP_SCREEN__
14: #define __PREP_SCREEN__
15: 
16: 
17: #include "../include/main.h"
18: 
19: #include "../Styles.hpp"
20: #include "OptionsScreen.hpp"
21: 
22: 
23: /**
24:  * @see: ../Styles.hpp
25:  * @see: ../gui.hpp
26:  *
27:  * final confirmation steps for auton
28:  * shows the initialization that will occur after the user clicks continue
29:  */
30: class PrepScreen : private Styles
31: {
32:     private:
33:         //screen
34:         lv_obj_t *prep_screen;
35: 
36:         //labels
37:         lv_obj_t *title_label;
38: 
39:         lv_obj_t *actions_label;
40: 
41:         //buttons
42:         lv_obj_t *btn_confirm;
43:         lv_obj_t *btn_back;
44: 
45:         //button labels
46:         lv_obj_t *btn_confirm_label;
47:         lv_obj_t *btn_back_label;
48: 
49: 
50:     public:
51:         PrepScreen();
52:         ~PrepScreen();
53: 
54:         static bool nextScreen;
55:         static bool confirm;
56: 
57:         //button functions
58: 
59:         /**
60:          * @param: lv_obj_t* btn -> button that called the funtion
61:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
62:          *
63:          * button callback function used to set variable so that gui continues
64:          * to the next stage
65:          */
66:         static lv_res_t btn_confirm_action(lv_obj_t *btn);
67: 
68: 
69:         /**
70:          * @param: lv_obj_t* btn -> button that called the funtion
71:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
72:          *
73:          * button callback function used to set variable so that gui goes
74:          * to the previous stage
75:          */
76:         static lv_res_t btn_back_action(lv_obj_t *btn);
77: 
78: 
79:         /**
80:          * @param: int auton -> auton number selected, used to set border color of gui based on side of color the auton is run on
81:          * @return: None
82:          *
83:          * @see: ../Structs.hpp
84:          * @see: ../Styles.hpp
85:          *
86:          * gunction used to get confirmation from user to continue to next stage
87:          * of the selection process
88:          */
89:         void getConfirmation( int auton );
90: 
91: };
92: #endif
```

```
 1: /**
 2:  * @file: ./RobotCode/src/lcdCode/AutonSelecton/PrepScreen.cpp
 3:  * @author: Aiden Carney
 4:  * @reviewed_on: 10/15/2019
 5:  * @reviewed_by: Aiden Carney
 6:  *
 7:  * @see: PrepScreen.hpp
 8:  *
 9:  * contains class methods seeking confirmation from user
10:  */
11:
12: #include <sstream>
13: #include <string>
14:
15: #include "../../../../include/main.h"
16: #include "../../../../include/api.h"
17:
18: #include "../../../Autons.hpp"
19: #include "../../controller/controller.hpp"
20: #include "PrepScreen.hpp"
21:
22:
23: bool PrepScreen::nextScreen = false;
24: bool PrepScreen::confirm = false;
25:
26:
27:
28: PrepScreen::PrepScreen()
29: {
30:     nextScreen = false;
31:     confirm = false;
32:
33:     prep_screen = lv_obj_create(NULL, NULL);
34:
35: //actions_label
36:     actions_label = lv_label_create(prep_screen, NULL);
37:     lv_obj_set_style(actions_label, &heading_text);
38:     lv_obj_set_width(actions_label, 300);
39:     lv_obj_set_height(actions_label, 160);
40:     lv_label_set_align(actions_label, LV_LABEL_ALIGN_LEFT);
41:     lv_label_set_text(actions_label, "actions");
42:
43:
44:
45: //confirm button
46:     //button
47:     btn_confirm = lv_btn_create(prep_screen, NULL);
48:     lv_btn_set_style(btn_confirm, LV_BTN_STYLE_REL, &toggle_btn_released);
49:     lv_btn_set_style(btn_confirm, LV_BTN_STYLE_PR, &toggle_btn_pressed);
50:     lv_btn_set_action(btn_confirm, LV_BTN_ACTION_CLICK, btn_confirm_action);
51:     lv_obj_set_width(btn_confirm, 300);
52:     lv_obj_set_height(btn_confirm, 25);
53:
54:     //label
55:     btn_confirm_label = lv_label_create(btn_confirm, NULL);
56:     lv_obj_set_style(btn_confirm_label, &heading_text);
57:     lv_label_set_text(btn_confirm_label, "Confirm");
58:
59: //back button
60:     //button
61:     btn_back = lv_btn_create(prep_screen, NULL);
62:     lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
63:     lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
64:     lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
65:     lv_obj_set_width(btn_back, 50);
66:     lv_obj_set_height(btn_back, 25);
67:
68:     //label
69:     btn_back_label = lv_label_create(btn_back, NULL);
70:     lv_obj_set_style(btn_back_label, &heading_text);
71:     lv_label_set_text(btn_back_label, "Back");
72:
73:
74: //title label
75:     title_label = lv_label_create(prep_screen, NULL);
76:     lv_obj_set_style(title_label, &heading_text);
77:     lv_obj_set_width(title_label, 300);
78:     lv_obj_set_height(title_label, 20);
79:     lv_label_set_align(title_label, LV_LABEL_ALIGN_LEFT);
80:     lv_label_set_text(title_label, "Auton");
81:
82:
83:     lv_obj_set_pos(btn_back, 40, 200);
84:     lv_obj_set_pos(btn_confirm, 100, 200);
85:     lv_obj_set_pos(title_label, 210, 20);
86:     lv_obj_set_pos(actions_label, 40, 50);
87: }
88:
89:
90:
91:
92: PrepScreen::~PrepScreen()
93: {
94:     lv_obj_del(btn_back_label);
95:     lv_obj_del(btn_confirm_label);
96:     lv_obj_del(title_label);
97:
98:     lv_obj_del(btn_back);
99:     lv_obj_del(btn_confirm);
100:
101:     lv_obj_del(prep_screen);
102: }
103:
104:
105:
106:
107: /**
108:  * sets nextScreen so that main loop will break and go to next stage
109:  */
110: lv_res_t PrepScreen::btn_confirm_action(lv_obj_t *btn)
111: {
112:     nextScreen = true;
113:     confirm = true;
```

```
114:
115:     return LV_RES_OK;
116:  }
117:
118:  /**
119:   * sets nextScreen so that main loop will break and go to the previous stage
120:   */
121:  lv_res_t PrepScreen::btn_back_action(lv_obj_t *btn)
122:  {
123:     nextScreen = true;
124:     confirm = false;
125:
126:     return LV_RES_OK;
127:  }
128:
129:
130:
131:  /**
132:   * runs loop where user can see what operations will be performed
133:   * loop breaks when user clicks the back or continue button
134:   * if user click the back button then the back flag is set
135:   */
136:  void PrepScreen::getConfirmation( int auton )
137:  {
138:     Controller controllers;
139:     Autons auton_data;
140:
141:     lv_label_set_text(title_label, auton_data.AUTONOMOUS_NAMES.at(auton));
142:
143:     //load screen
144:     lv_scr_load(prep_screen);
145:
146:     //set color of border
147:     std::string color = auton_data.AUTONOMOUS_COLORS.at(auton);
148:     if (color == "blue")
149:     {
150:        gray.body.border.color = BLUE_BORDER;
151:     }
152:     else if (color == "red")
153:     {
154:        gray.body.border.color = RED_BORDER;
155:     }
156:     else
157:     {
158:        gray.body.border.color = BG;
159:     }
160:
161:     lv_obj_set_style(prep_screen, &gray);
162:
163:     nextScreen = false;
164:
165:
166:     std::string label =
167:        "Initialize and Calibrate Gyro\n"
168:        "Initialize Other Sensors\n"
169:        "Initialize Motors\n"
170:        "Zero Motor Encoders\n"
171:        "Initialize Controllers\n";
172:     if ( OptionsScreen::cnfg.record )
173:     {
174:        label = label + "Start Recording Thread";
175:     }
176:
177:     //cast std::string to const char* and set text
178:     std::ostringstream text;
179:     text << label;
180:     lv_label_set_text(actions_label, text.str().c_str());
181:
182:     pros::delay( 100 );  //add delay so that button press from previous stage does not register
183:
184:     while ( !(nextScreen) )
185:     {
186:        if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_A) )
187:        {
188:           btn_confirm_action( NULL );
189:           pros::delay(200);
190:        }
191:        else if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_B) )
192:        {
193:           btn_back_action( NULL );
194:           pros::delay(200);
195:        }
196:        pros::delay(20);
197:     }
198:
199:  }
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/AutonSelecton/SelectionScreen.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   * TODO: add ability for controller to make selections
7:   *
8:   * contains first stage of auton selection--selecting the auton number
9:   *
10:  */
11:
12: #ifndef __SELECTIONSCREEN_HPP__
13: #define __SELECTIONSCREEN_HPP__
14:
15:
16: #include "../../../../include/main.h"
17:
18: #include "../Styles.hpp"
19:
20:
21:
22:
23: /**
24:  * @see: ../Styles.hpp
25:  *
26:  * contains methods for going through each auton and providing description
27:  * for user
28:  */
29: class SelectionScreen : private Styles
30: {
31:     private:
32:
33:         //labels
34:         lv_obj_t *title_label;
35:         lv_obj_t *description_label;
36:         lv_obj_t *auton_number_label;
37:
38:         //buttons
39:         lv_obj_t *btn_right;
40:         lv_obj_t *btn_left;
41:         lv_obj_t *btn_select;
42:
43:         //button labels
44:         lv_obj_t *btn_right_label;
45:         lv_obj_t *btn_left_label;
46:         lv_obj_t *btn_select_label;
47:
48:
49:     public:
50:         //screens
51:         static lv_obj_t *selection_screen;
52:         //variables
53:         static int auton_choice;
54:         static int final_choice;
55:         static bool update;
56:
57:         //button action functions
58:
59:         /**
60:          * @param: lv_obj_t* btn -> button that called the funtion
61:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
62:          *
63:          * button callback function used to set variable so that gui moves
64:          * autons to the right (increasing auton number by one and looping at end)
65:          */
66:         static lv_res_t btn_right_action(lv_obj_t *btn);
67:
68:
69:         /**
70:          * @param: lv_obj_t* btn -> button that called the funtion
71:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
72:          *
73:          * button callback function used to set variable so that gui moves
74:          * autons to the left (decreasing auton number by one and looping at end)
75:          */
76:         static lv_res_t btn_left_action(lv_obj_t *btn);
77:
78:
79:         /**
80:          * @param: lv_obj_t* btn -> button that called the funtion
81:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
82:          *
83:          * button callback function used to select the auton so that the gui continues
84:          * to the next stage
85:          */
86:         static lv_res_t btn_select_action(lv_obj_t *btn);
87:
88:         SelectionScreen();
89:         ~SelectionScreen();
90:
91:
92:
93:         /**
94:          * @return: None
95:          *
96:          * @see: int selectAuton()
97:          * @see: ../Structs.hpp
98:          *
99:          * sets the description, color, title, and auton number on the screen
100:         * used to update the auton selection based on the current number
101:         */
102:        void showSelection();
103:
104:
105:        /**
106:         * @param: int auton -> auton number to start the screen at
107:         * @return: int -> auton number that the screen was on when the user hit the select button
108:         *
109:         * loops through waiting for the auton number to change to update the screen
110:         */
111:        int selectAuton( int auton );
112:
113:
```

```
114:  };
115:
116:
117:
118:  #endif
```

```
 1: /**
 2:  * @file: ./RobotCode/src/lcdCode/AutonSelecton/SelectionScreen.cpp
 3:  * @author: Aiden Carney
 4:  * @reviewed_on: 10/15/2019
 5:  * @reviewed_by: Aiden Carney
 6:  *
 7:  * @see: SelectionScreen.hpp
 8:  *
 9:  * contains methods for class that give user the ability to select an auton number
10:  */
11:
12: #include <sstream>
13: #include <string>
14:
15: #include "../../../../include/main.h"
16:
17: #include "../../../Autons.hpp"
18: #include "SelectionScreen.hpp"
19: #include "../../controller/controller.hpp"
20:
21: //init static vars
22: lv_obj_t *SelectionScreen::selection_screen;
23: int SelectionScreen::auton_choice = 1;
24: int SelectionScreen::final_choice = 0;
25: bool SelectionScreen::update = false;
26:
27:
28: //constructor
29: SelectionScreen::SelectionScreen()
30:
31: {
32:     auton_choice = 1;
33:     final_choice = 0;
34:     update = false;
35:
36:     selection_screen = lv_obj_create(NULL, NULL);
37:
38:
39:     //init buttons and labels
40:     title_label = lv_label_create(selection_screen, NULL);
41:     description_label = lv_label_create(selection_screen, NULL);
42:     auton_number_label = lv_label_create(selection_screen, NULL);
43:
44:     btn_right = lv_btn_create(selection_screen, NULL);
45:     btn_left = lv_btn_create(selection_screen, NULL);
46:     btn_select = lv_btn_create(selection_screen, NULL);
47:
48:     btn_right_label = lv_label_create(btn_right, NULL);
49:     btn_left_label = lv_label_create(btn_left, NULL);
50:     btn_select_label = lv_label_create(btn_select, NULL);
51:
52:
53:     //sets style for widgets
54:     lv_obj_set_style(selection_screen, &gray);
55:
56:     lv_btn_set_style(btn_right, LV_BTN_STYLE_REL, &toggle_btn_released);
57:     lv_btn_set_style(btn_left, LV_BTN_STYLE_REL, &toggle_btn_released);
58:     lv_btn_set_style(btn_select, LV_BTN_STYLE_REL, &toggle_btn_released);
59:
60:     lv_btn_set_style(btn_right, LV_BTN_STYLE_PR, &toggle_btn_pressed);
61:     lv_btn_set_style(btn_left, LV_BTN_STYLE_PR, &toggle_btn_pressed);
62:     lv_btn_set_style(btn_select, LV_BTN_STYLE_PR, &toggle_btn_pressed);
63:
64:     lv_obj_set_style(btn_right_label, &heading_text);
65:     lv_obj_set_style(btn_left_label, &heading_text);
66:     lv_obj_set_style(btn_select_label, &heading_text);
67:
68:     lv_label_set_style(title_label, &heading_text);
69:     lv_label_set_style(description_label, &heading_text);
70:     lv_label_set_style(auton_number_label, &heading_text);
71:
72:     lv_label_set_long_mode(description_label, LV_LABEL_LONG_BREAK);
73:     lv_label_set_align(auton_number_label, LV_LABEL_ALIGN_CENTER);
74:
75:     lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
76:     lv_label_set_align(description_label, LV_LABEL_ALIGN_CENTER);
77:
78:
79:     //set size of widgets
80:     lv_obj_set_width(btn_right, 80);
81:     lv_obj_set_width(btn_left, 80);
82:     lv_obj_set_width(btn_select, 120);
83:     lv_obj_set_width(auton_number_label, 40);
84:     lv_obj_set_width(title_label, 400);
85:     lv_obj_set_width(description_label, 400);
86:
87:     lv_obj_set_height(btn_right, 80);
88:     lv_obj_set_height(btn_left, 80);
89:     lv_obj_set_height(btn_select, 40);
90:     lv_obj_set_height(auton_number_label, 40);
91:     lv_obj_set_height(title_label, 30);
92:     lv_obj_set_height(description_label, 80);
93:
94:
95:     //set default text and move widgets to start location
96:     lv_label_set_text(btn_right_label, SYMBOL_RIGHT);
97:     lv_label_set_text(btn_left_label, SYMBOL_LEFT);
98:     lv_label_set_text(btn_select_label, "Select");
99:
100:    lv_obj_set_pos(btn_right, 390, 150);
101:    lv_obj_set_pos(btn_left, 10, 150);
102:    lv_obj_set_pos(btn_select, 180, 180);
103:    lv_obj_set_pos(auton_number_label, 440, 20);
104:    lv_obj_set_pos(title_label, 210, 20);
105:    lv_obj_set_pos(description_label, 40, 60);
106:
107:
108:    //set action for buttons
109:    lv_btn_set_action(btn_right, LV_BTN_ACTION_CLICK, btn_right_action);
110:    lv_btn_set_action(btn_left, LV_BTN_ACTION_CLICK, btn_left_action);
111:    lv_btn_set_action(btn_select, LV_BTN_ACTION_CLICK, btn_select_action);
112:
113: }
```

```
114:
115:
116: //destructor
117: SelectionScreen::~SelectionScreen()
118: {
119:     lv_obj_del(auton_number_label);
120:     lv_obj_del(title_label);
121:     lv_obj_del(description_label);
122:     lv_obj_del(btn_right);
123:     lv_obj_del(btn_left);
124:     lv_obj_del(btn_select);
125:
126:     lv_obj_del(selection_screen);
127: }
128:
129:
130:
131: //button action functions
132:
133: /**
134:  * called when left button is clicked
135:  * decrements auton_choice and loops it back in range if not in range
136:  */
137: lv_res_t SelectionScreen::btn_left_action(lv_obj_t *btn)
138: {
139:     Autons auton_data;
140:
141:     auton_choice -= 1;
142:     if (auton_choice < 1)
143:     {
144:         auton_choice = auton_data.AUTONOMOUS_NAMES.size();
145:     }
146:
147:     update = true;
148:
149:     return LV_RES_OK;
150: }
151:
152:
153: /**
154:  * called when left button is clicked
155:  * increments auton_choice and loops it back in range if not in range
156:  */
157: lv_res_t SelectionScreen::btn_right_action(lv_obj_t *btn)
158: {
159:     std::cout << "function called\n";
160:     Autons auton_data;
161:
162:     auton_choice += 1;
163:     if (auton_choice > auton_data.AUTONOMOUS_NAMES.size())
164:     {
165:         auton_choice = 1;
166:     }
167:
168:     update = true;
169:
170:     return LV_RES_OK;
171: }
172:
173:
174: /**
175:  * breaks main loop by setting the final auton choice so that gui continues
176:  */
177: lv_res_t SelectionScreen::btn_select_action(lv_obj_t *btn)
178: {
179:     final_choice = auton_choice;
180:     update = true;
181:
182:     return LV_RES_OK;
183: }
184:
185:
186:
187:
188: //other functions
189:
190: /**
191:  * updates background color by looking at std::unordered_map
192:  * updates auton number label
193:  * waits for there to be an update to be implemented by the buttons before exiting
194:  */
195: void SelectionScreen::showSelection()
196: {
197:     Controller controllers;
198:     Autons auton_data;
199:
200:     lv_label_set_text(title_label, auton_data.AUTONOMOUS_NAMES.at(auton_choice));
201:     lv_label_set_text(description_label, auton_data.AUTONOMOUS_DESCRIPTIONS.at(auton_choice));
202:
203:
204:     //get color
205:     std::string color = auton_data.AUTONOMOUS_COLORS.at(auton_choice);
206:     if (color == "blue")
207:     {
208:         gray.body.border.color = BLUE_BORDER;
209:     }
210:     else if (color == "red")
211:     {
212:         gray.body.border.color = RED_BORDER;
213:     }
214:     else
215:     {
216:         gray.body.border.color = BG;
217:     }
218:
219:     lv_obj_set_style(selection_screen, &gray); //update background
220:
221:     //cast int of auton choice to string
222:     std::string str_auton_choice;
223:     str_auton_choice = std::to_string(auton_choice);
224:     lv_label_set_text(auton_number_label, str_auton_choice.c_str());
225:
226:
```

```
227:     controllers.master.print( 0, 0, "                    ");
228:     pros::delay(50);
229:     controllers.master.print( 0, 0, auton_data.AUTONOMOUS_NAMES.at(auton_choice) );
230:
231:     while ( !(update) && !(final_choice) ) //waits for screen to change
232:                                 //so that time is not wasted
233:     {
234:       //allow controller to press the buttons as well
235:       if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_R1) )
236:       {
237:         btn_right_action( NULL );
238:         pros::delay(200);
239:       }
240:       else if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_L1) )
241:       {
242:         btn_left_action( NULL );
243:         pros::delay(200);
244:       }
245:       else if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_A) )
246:       {
247:         btn_select_action( NULL );
248:         pros::delay(200);
249:       }
250:       pros::delay(100);
251:     }
252:     update = false;
253:   }
254:
255:
256:
257:
258:   /**
259:    * waits in a loop for there to be an update to the gui implemented by
260:    * button callback functions
261:    * in the loop, the gui is updated until a final selection is made
262:    * everytime there is a change ie. when a button is clicked
263:    */
264:   int SelectionScreen::selectAuton( int auton )
265:   {
266:     auton_choice = auton;
267:     final_choice = 0;
268:     update = false;
269:
270:     //load screen
271:     lv_scr_load(selection_screen);
272:
273:     while ( !(final_choice) ) //waits for user to select an auton
274:     {                   //before going to next screen
275:       showSelection(); //showSelection contains delay
276:     }
277:
278:     gray.body.border.color = BG; //reset gray style
279:
280:     return final_choice;
281:
282:   }
```

```
 1: /**
 2:  * @file: ./RobotCode/src/lcdCode/Debug/Debug.hpp
 3:  * @author: Aiden Carney
 4:  * @reviewed_on: 10/15/2019
 5:  * @reviewed_by: Aiden Carney
 6:  *
 7:  * gives user the option to visit debugger tabs by selecting an option from a button
 8:  * matrix
 9:  */
10:
11: #ifndef __DEBUG_HPP__
12: #define __DEBUG_HPP__
13:
14: #include "BatteryDebug.hpp"
15: #include "ControllerDebug.hpp"
16: #include "FieldControlDebug.hpp"
17: #include "InternalMotorDebug.hpp"
18: #include "MotorsDebug.hpp"
19: #include "SensorsDebug.hpp"
20: #include "TitleScreen.hpp"
21: #include "Wiring.hpp"
22:
23:
24:
25: /**
26:  * @return: None
27:  *
28:  * @see: TitleScreen.hpp
29:  *
30:  * loads screens and switches the debugger option based on a what is clicked from
31:  * a button matrix
32:  */
33: void debug();
34:
35:
36: #endif
```

```
 1: /**
 2:  * @file: ./RobotCode/src/lcdCode/Debug/Debug.cpp
 3:  * @author: Aiden Carney
 4:  * @reviewed_on: 10/15/2019
 5:  * @reviewed_by: Aiden Carney
 6:  *
 7:  * @see: Debug.hpp
 8:  *
 9:  * contains function for selecting debug screen
10:  *
11:  */
12:
13: #include "../../../../include/main.h"
14: #include "../../../../include/api.h"
15:
16: #include "Debug.hpp"
17:
18:
19:
20: /**
21:  * loads all screens at beginning
22:  * when on titlescreen a tab number is selected and a switch statement is used
23:  * to let a tab take over
24:  */
25: void debug()
26: {
27:     bool cont = true;
28:
29:     TitleScreen dbg1;
30:     MotorsDebug dbgM;
31:     SensorsDebug dbgS;
32:     ControllerDebug dbgC;
33:     BatteryDebug dbgB;
34:     FieldControlDebug dbgF;
35:     Wiring dbgW;
36:     InternalMotorDebug dbgP;
37:
38:
39:     while ( cont )
40:     {
41:         dbg1.chooseOption();
42:         if ( dbg1.option == -1 ) //-1 means go back
43:         {
44:             cont = false;
45:             break;
46:         }
47:
48:         switch (dbg1.option) //go to selected debug screen
49:         {
50:             case 1:
51:                 dbgM.debug();
52:                 break;
53:
54:             case 2:
55:                 dbgS.debug();
56:                 break;
57:
58:             case 3:
59:                 dbgC.debug();
60:                 break;
61:
62:             case 4:
63:                 dbgB.debug();
64:                 break;
65:
66:             case 5:
67:                 dbgF.debug();
68:                 break;
69:
70:             case 6:
71:                 dbgW.debug();
72:                 break;
73:             case 7:
74:                 dbgP.debug();
75:                 break;
76:         }
77:
78:     }
79:
80: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/Debug/BatteryDebug.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/16/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains class for debugging the battery
8:   */
9:
10: #ifndef __BATTERYDEBUG_HPP__
11: #define __BATTERYDEBUG_HPP__
12:
13:
14: #include "../../../../include/main.h"
15:
16: #include "../Styles.hpp"
17:
18:
19: /**
20:  * @see: ../Styles.hpp
21:  *
22:  * contains methods that show user data about the battery
23:  */
24: class BatteryDebug : private Styles
25: {
26:     private:
27:         lv_obj_t *battery_screen;
28:         lv_obj_t *title_label;
29:
30:         lv_obj_t *labels_label;
31:         lv_obj_t *info_label;
32:
33:
34:         //back button
35:         lv_obj_t *btn_back;
36:         lv_obj_t *btn_back_label;
37:
38:         /**
39:          * @param: lv_obj_t* btn -> button that called the funtion
40:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
41:          *
42:          * button callback function used to set cont to false meaning the
43:          * user wants to go to the title screen
44:          */
45:         static lv_res_t btn_back_action(lv_obj_t *btn);
46:
47:     public:
48:         static bool cont;
49:
50:         BatteryDebug();
51:         ~BatteryDebug();
52:
53:
54:         /**
55:          * @return: None
56:          *
57:          * allows user to see information about the state of field control
58:          */
59:         void debug();
60:
61: };
62:
63:
64:
65:
66: #endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/BatteryDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * @see BatteryDebug.hpp
 *
 * contains implementation for class for debugging the battery
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "BatteryDebug.hpp"


bool BatteryDebug::cont = true;

BatteryDebug::BatteryDebug()
{
    cont = true;

//screen
    battery_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(battery_screen, &gray);

//init back button
    //button
    btn_back = lv_btn_create(battery_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init title label
    title_label = lv_label_create(battery_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 440);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Battery - Debug");

//init headers label
    labels_label = lv_label_create(battery_screen, NULL);
    lv_label_set_style(labels_label, &subheading_text);
    lv_obj_set_width(labels_label, 220);
    lv_obj_set_height(labels_label, 200);
    lv_label_set_align(labels_label, LV_LABEL_ALIGN_LEFT);

    std::string labels_label_text = (
        "battery percentage\n"
        "current\n"
        "voltage\n"
        "temperature"
    );

    lv_label_set_text(labels_label, labels_label_text.c_str());

//init values label
    info_label = lv_label_create(battery_screen, NULL);
    lv_label_set_style(info_label, &subheading_text);
    lv_obj_set_width(info_label, 220);
    lv_obj_set_height(info_label, 200);
    lv_label_set_align(info_label, LV_LABEL_ALIGN_LEFT);

    std::string info_label_text = (
        "None\n"
        "None\n"
        "None\n"
        "None"
    );

    lv_label_set_text(info_label, info_label_text.c_str());

//set positions
    lv_obj_set_pos(btn_back, 30, 210);

    lv_obj_align(title_label, battery_screen, LV_ALIGN_IN_TOP_MID, 0, 10);

    lv_obj_set_pos(labels_label, 20, 40);
    lv_obj_set_pos(info_label, 360, 40);


}


BatteryDebug::~BatteryDebug()
{
    lv_obj_del(battery_screen);
}



/**
 * sets cont to false to break main loop so main function returns
 */
lv_res_t BatteryDebug::btn_back_action(lv_obj_t *btn)
{
    cont = false;
    return LV_RES_OK;
}
```

```
114:  /**
115:   * main loop that updates the battery information
116:   */
117:  void BatteryDebug::debug()
118:  {
119:      cont = true;
120:
121:      lv_scr_load(battery_screen);
122:
123:      while ( cont )
124:      {
125:          std::string info_label_text = (
126:              std::to_string(pros::battery::get_capacity()) + "%\n"
127:              + std::to_string(pros::battery::get_current()) + "\n"
128:              + std::to_string(pros::battery::get_voltage()) + "\n"
129:              + std::to_string(pros::battery::get_temperature()) + "\n"
130:          );
131:
132:          lv_label_set_text(info_label, info_label_text.c_str());
133:
134:
135:          pros::delay(100);
136:      }
137:  }
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/ControllerDebug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 * TODO:
 *
 * contains classes to show data about controller on gui
 */

#ifndef __CONTROLLERDEBUG_HPP__
#define __CONTROLLERDEBUG_HPP__

#include <unordered_map>

#include "../../../../include/main.h"

#include "../Styles.hpp"

//user defines

//sets size of container
#define CONTROLLER_CONTAINER_WIDTH 440
#define CONTROLLER_CONTAINER_HEIGHT 120

//Base classes
//base classes are the tabs that will be loaded by the derived class
//this makes it easy to add new tabs while keeping the amount that has
//to go in one class to a minimum, especially since lvgl is not light



/**
 * @see: ../Styels.hpp
 * @see: ../../controller/controller.hpp
 *
 * contains general information about controllers and also allow
 * to test communication by sending rumbles and test strings
 */
class GeneralControllerDebug : virtual Styles
{
    private:
        lv_obj_t *container;

        lv_obj_t *controller_column;
        lv_obj_t *connected_column;
        lv_obj_t *capacity_column;
        lv_obj_t *level_column;



        lv_obj_t *btn_test_string;
        lv_obj_t *btn_test_string_label;

        /**
         * @param: lv_obj_t* btn -> button that called the function
         * @return: lv_res_t -> return LV_RES_OK because object still exists
         * TODO: sometimes string does not send to the write location, fix
         *
         * sends a simple string to the controller lcd to test where output is
         * when the user clicks a button
         */
        static lv_res_t btn_test_string_action(lv_obj_t *btn);



        lv_obj_t *btn_clear_scr;
        lv_obj_t *btn_clear_scr_label;

        /**
         * @param: lv_obj_t* btn -> button that called the function
         * @return: lv_res_t -> return LV_RES_OK because object still exists
         * TODO: sometimes doesn't actually work
         *
         * clears any output on the lcd controller when user clicks a button
         */
        static lv_res_t btn_clear_scr_action(lv_obj_t *btn);



        lv_obj_t *btn_test_rumble;
        lv_obj_t *btn_test_rumble_label;

        /**
         * @param: lv_obj_t* btn -> button that called the function
         * @return: lv_res_t -> return LV_RES_OK because object still exists
         *
         * tells controller to rumble
         */
        static lv_res_t btn_test_rumble_action(lv_obj_t *btn);



        lv_obj_t *controls_info;
        lv_obj_t *motor2_info;

    protected:
        /**
         * @return: None
         *
         * @see: ../../controller/controller.hpp
         *
         * updates data for the controllers such as connected or not, and battery
         */
        void update_general_info();

    public:
        GeneralControllerDebug();
        virtual ~GeneralControllerDebug();

        /**
         * @param: lv_obj_t* parent -> new parent for the main container
```

```
114:          * @return: None
115:          * TODO: depracate and fix method of inheritence, current method is not implemented well
116:          *
117:          * changes parent of containter
118:          */
119:         void GeneralControllerDebugInit(lv_obj_t *parent);
120:
121:  };
122:
123:
124:
125:
126:  /**
127:   * @see: ../Styels.hpp
128:   * @see: ../../controller/controller.hpp
129:   *
130:   * generic class for showing the functions or values of the controller for each button
131:   */
132:  class ControllerTab : virtual Styles
133:  {
134:      private:
135:         lv_obj_t *container;
136:
137:         //separated into two columns because LCD is not big enough
138:         //column one widgets
139:         lv_obj_t *button_names_one;
140:         lv_obj_t *button_col_one;
141:
142:         //column two widgets
143:         lv_obj_t *button_names_two;
144:         lv_obj_t *button_col_two;
145:
146:      public:
147:         ControllerTab(lv_obj_t *parent);
148:         virtual ~ControllerTab();
149:
150:
151:         /**
152:          * @param: pros::controller_id_e_t controller -> controller that the tab is looking at
153:          * @param: bool showing_values -> bool for if values should be shown or not
154:          * @return: None
155:          * TODO: make controller a member so that controller does not have to be a parameter
156:          *
157:          * shows either the values or the function the controller calls on the gui
158:          */
159:         void update(pros::controller_id_e_t controller, bool showing_values);
160:  };
161:
162:
163:
164:
165:
166:  //derived class
167:
168:
169:  /**
170:   * @see: ../Styles.hpp
171:   *
172:   * tab for showing data about controllers
173:   */
174:  class ControllerDebug :
175:      virtual private Styles,
176:      private GeneralControllerDebug
177:  {
178:      private:
179:         static bool showing_values;
180:
181:         //screen
182:         lv_obj_t *controller_debug_screen;
183:
184:         //title label
185:         lv_obj_t *title_label;
186:
187:
188:
189:         lv_obj_t *btn_back;
190:         lv_obj_t *btn_back_label;
191:
192:         /**
193:          * @param: lv_obj_t* btn -> button that called the funtion
194:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
195:          *
196:          * button callback function used to set cont to false meaning the
197:          * user wants to go to the title screen
198:          */
199:         static lv_res_t btn_back_action(lv_obj_t *btn);
200:
201:
202:
203:         lv_obj_t *btn_show_values;
204:         static lv_obj_t *btn_show_values_label;
205:
206:         /**
207:          * @param: lv_obj_t* btn -> button that called the funtion
208:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
209:          *
210:          * button callback function switch between showing the functions or values of
211:          * for the controller
212:          */
213:         static lv_res_t btn_show_values_action(lv_obj_t *btn);
214:
215:
216:
217:         static lv_obj_t *tabview; //tabview object
218:
219:         //individual tabs
220:         //content will come from base classes
221:         lv_obj_t *general_tab;
222:         lv_obj_t *master_tab;
223:         lv_obj_t *partner_tab;
224:
225:
226:      public:
```

```
227:        ControllerDebug();
228:        ~ControllerDebug();
229:
230:        static bool cont; //checks whether to keep letting user
231:                          //cycle through tabs
232:
233:
234:        /**
235:        * @return: None
236:        *
237:        * allows user to see information about the controller
238:        */
239:        void debug();
240:    };
241:
242:
243:
244:
245:    #endif
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/ControllerDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: ControllerDebug.hpp
 *
 * contains implementation for classes that show information about the controller
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include <unordered_map>

#include "../Styles.hpp"
#include "../../controller/controller.hpp"
#include "ControllerDebug.hpp"


//declare static members of all classes
bool ControllerDebug::showing_values = 0;
bool ControllerDebug::cont = true;
lv_obj_t *ControllerDebug::tabview;
lv_obj_t *ControllerDebug::btn_show_values_label;



GeneralControllerDebug::GeneralControllerDebug()
{
//init container
    container = lv_cont_create(lv_scr_act(), NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, CONTROLLER_CONTAINER_WIDTH);
    lv_obj_set_height(container, CONTROLLER_CONTAINER_HEIGHT);

//default text for each column
    std::string text1 = (
        "Controller\n"
        "Master\n"
        "Partner"
    );

    std::string text2 = (
        "Connected\n"
        "no\n"
        "no"
    );

    std::string text3 = (
        "Battery Capacity\n"
        "0\n"
        "0"
    );

    std::string text4 = (
        "Battery Percent\n"
        "0\n"
        "0"
    );

//init controller column label
    controller_column = lv_label_create(container, NULL);
    lv_obj_set_style(controller_column, &toggle_tabbtn_pressed);
    lv_obj_set_width(controller_column, (CONTROLLER_CONTAINER_WIDTH / 4));
    lv_obj_set_height(controller_column, CONTROLLER_CONTAINER_HEIGHT);
    lv_label_set_align(controller_column, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(controller_column, text1.c_str());

//init connected column label
    connected_column = lv_label_create(container, NULL);
    lv_obj_set_style(connected_column, &toggle_tabbtn_pressed);
    lv_obj_set_width(connected_column, (CONTROLLER_CONTAINER_WIDTH / 4));
    lv_obj_set_height(connected_column, CONTROLLER_CONTAINER_HEIGHT);
    lv_label_set_align(connected_column, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(connected_column, text2.c_str());

//init capacity column label
    capacity_column = lv_label_create(container, NULL);
    lv_obj_set_style(capacity_column, &toggle_tabbtn_pressed);
    lv_obj_set_width(capacity_column, (CONTROLLER_CONTAINER_WIDTH / 4));
    lv_obj_set_height(capacity_column, CONTROLLER_CONTAINER_HEIGHT);
    lv_label_set_align(capacity_column, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(capacity_column, text3.c_str());

//init battery percentage column label
    level_column = lv_label_create(container, NULL);
    lv_obj_set_style(level_column, &toggle_tabbtn_pressed);
    lv_obj_set_width(level_column, (CONTROLLER_CONTAINER_WIDTH / 4));
    lv_obj_set_height(level_column, CONTROLLER_CONTAINER_HEIGHT);
    lv_label_set_align(level_column, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(level_column, text4.c_str());


//init send test string button
    //button
    btn_test_string = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_test_string, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_test_string, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_test_string, LV_BTN_ACTION_CLICK, btn_test_string_action);
    lv_obj_set_width(btn_test_string, 130);
    lv_obj_set_height(btn_test_string, 25);

    //label
    btn_test_string_label = lv_label_create(btn_test_string, NULL);
    lv_obj_set_style(btn_test_string_label, &subheading_text);
    lv_label_set_text(btn_test_string_label, "Send Test String");


//init clear screen button
```

```
114:     //button
115:     btn_clear_scr = lv_btn_create(container, NULL);
116:     lv_btn_set_style(btn_clear_scr, LV_BTN_STYLE_REL, &toggle_btn_released);
117:     lv_btn_set_style(btn_clear_scr, LV_BTN_STYLE_PR, &toggle_btn_pressed);
118:     lv_btn_set_action(btn_clear_scr, LV_BTN_ACTION_CLICK, btn_clear_scr_action);
119:     lv_obj_set_width(btn_clear_scr, 130);
120:     lv_obj_set_height(btn_clear_scr, 25);
121:
122:     //label
123:     btn_clear_scr_label = lv_label_create(btn_clear_scr, NULL);
124:     lv_obj_set_style(btn_clear_scr_label, &subheading_text);
125:     lv_label_set_text(btn_clear_scr_label, "Clear Screen");
126:
127:
128:   //init send test rumble button
129:     //button
130:     btn_test_rumble = lv_btn_create(container, NULL);
131:     lv_btn_set_style(btn_test_rumble, LV_BTN_STYLE_REL, &toggle_btn_released);
132:     lv_btn_set_style(btn_test_rumble, LV_BTN_STYLE_PR, &toggle_btn_pressed);
133:     lv_btn_set_action(btn_test_rumble, LV_BTN_ACTION_CLICK, btn_test_rumble_action);
134:     lv_obj_set_width(btn_test_rumble, 130);
135:     lv_obj_set_height(btn_test_rumble, 25);
136:
137:     //label
138:     btn_test_rumble_label = lv_label_create(btn_test_rumble, NULL);
139:     lv_obj_set_style(btn_test_rumble_label, &subheading_text);
140:     lv_label_set_text(btn_test_rumble_label, "Send Test Rumble");
141:
142:
143:   //set positions relative to container
144:     lv_obj_align(controller_column, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
145:     lv_obj_align(connected_column, container, LV_ALIGN_IN_TOP_MID, -80, 10);
146:     lv_obj_align(capacity_column, container, LV_ALIGN_IN_TOP_MID, 15, 10);
147:     lv_obj_align(level_column, container, LV_ALIGN_IN_TOP_RIGHT, -30, 10);
148:
149:
150:     lv_obj_align(btn_test_string, container, LV_ALIGN_IN_BOTTOM_LEFT, 20, -30);
151:     lv_obj_align(btn_clear_scr, container, LV_ALIGN_IN_BOTTOM_MID, 0, -30);
152:     lv_obj_align(btn_test_rumble, container, LV_ALIGN_IN_BOTTOM_RIGHT, -20, -30);
153:
154:   }
155:
156:
157:   GeneralControllerDebug::~GeneralControllerDebug()
158:   {
159:
160:   }
161:
162:
163:
164:   /**
165:    * sends test string to controller
166:    */
167:   lv_res_t GeneralControllerDebug::btn_test_string_action(lv_obj_t *btn)
168:   {
169:     Controller::master.print(0, 0, "This is a test message");
170:     Controller::partner.print(0, 0, "This is a test message");
171:     return LV_RES_OK;
172:   }
173:
174:
175:   /**
176:    * clears the screen on the controller
177:    */
178:   lv_res_t GeneralControllerDebug::btn_clear_scr_action(lv_obj_t *btn)
179:   {
180:     Controller::master.print(0, 0, "");
181:     Controller::partner.print(0, 0, "");
182:     Controller::master.clear_line(0);
183:     Controller::partner.clear_line(0);
184:     return LV_RES_OK;
185:   }
186:
187:
188:   /**
189:    * sends a test rumble to the controller
190:    */
191:   lv_res_t GeneralControllerDebug::btn_test_rumble_action(lv_obj_t *btn)
192:   {
193:     Controller::master.rumble(". - - -");
194:     Controller::partner.rumble(". - - -");
195:     return LV_RES_OK;
196:   }
197:
198:
199:
200:
201:   /**
202:    * updates data on the tab
203:    */
204:   void GeneralControllerDebug::update_general_info()
205:   {
206:     std::string text1 = (
207:         "Controller\n"
208:         "Master\n"
209:         "Partner"
210:     );
211:
212:     std::string master_text = "no";
213:     std::string partner_text = "no";
214:     if ( Controller::master.is_connected() )
215:     {
216:         master_text = "yes";
217:     }
218:     if ( Controller::partner.is_connected() )
219:     {
220:         partner_text = "yes";
221:     }
222:
223:     std::string text2 = (
224:         "Connected\n"
225:         + master_text + "\n"
226:         + partner_text
```

```
227:     );
228:
229:
230:     std::string text3 = (
231:         "Battery Capacity\n"
232:         + std::to_string(Controller::master.get_battery_level()) + "\n"
233:         + std::to_string(Controller::partner.get_battery_level())
234:     );
235:
236:     std::string text4 = (
237:         "Battery Percent\n"
238:         + std::to_string(Controller::master.get_battery_capacity()) + "\n"
239:         + std::to_string(Controller::partner.get_battery_capacity())
240:     );
241:
242:     lv_label_set_text(controller_column, text1.c_str());
243:     lv_label_set_text(connected_column, text2.c_str());
244:     lv_label_set_text(capacity_column, text3.c_str());
245:     lv_label_set_text(level_column, text4.c_str());
246: }
247:
248:
249:
250: /**
251:  * changes parent of all objects
252:  */
253: void GeneralControllerDebug::GeneralControllerDebugInit(lv_obj_t *parent)
254: {
255:     //sets parent of container to pointer of new parent
256:     //this is to allow seperation of tabs into seperate classes
257:     //reduce the quantity in one class and to allow for ease of adding
258:     //new or different tabs
259:
260:     lv_obj_set_parent(container, parent);
261:
262: }
263:
264:
265:
266:
267: ControllerTab::ControllerTab(lv_obj_t *parent)
268: {
269: //init container
270:     container = lv_cont_create(parent, NULL);
271:     lv_cont_set_fit(container, false, false);
272:     lv_obj_set_style(container, &gray);
273:     lv_cont_set_fit(container, false, false);
274:     lv_obj_set_width(container, CONTROLLER_CONTAINER_WIDTH);
275:     lv_obj_set_height(container, CONTROLLER_CONTAINER_HEIGHT);
276:
277: //text for names
278:     std::string ctrl_col1 = (
279:         "Analog Left X\n"
280:         "Analog Left Y\n"
281:         "Analog Right X\n"
282:         "Analog Right Y\n"
283:         "Digital L1\n"
284:         "Digital L2\n"
285:         "Digital R1\n"
286:         "Digital R2"
287:     );
288:
289:     std::string ctrl_col2 = (
290:         "Digital Up\n"
291:         "Digital Down\n"
292:         "Digital Left\n"
293:         "Digital Right\n"
294:         "Digital X\n"
295:         "Digital B\n"
296:         "Digital Y\n"
297:         "Digital A\n"
298:     );
299:
300: //column one button names
301:     button_names_one = lv_label_create(container, NULL);
302:     lv_label_set_style(button_names_one, &toggle_tabbtn_pressed);
303:     lv_obj_set_width(button_names_one, CONTROLLER_CONTAINER_WIDTH / 4);
304:     lv_obj_set_height(button_names_one, 20);
305:     lv_label_set_align(button_names_one, LV_LABEL_ALIGN_LEFT);
306:     lv_label_set_text(button_names_one, ctrl_col1.c_str());
307:
308:
309: //column two button names
310:     button_names_two = lv_label_create(container, NULL);
311:     lv_label_set_style(button_names_two, &toggle_tabbtn_pressed);
312:     lv_obj_set_width(button_names_two, CONTROLLER_CONTAINER_WIDTH / 4);
313:     lv_obj_set_height(button_names_two, 20);
314:     lv_label_set_align(button_names_two, LV_LABEL_ALIGN_LEFT);
315:     lv_label_set_text(button_names_two, ctrl_col2.c_str());
316:
317:
318: //column one second part that contains function or values
319:     button_col_one = lv_label_create(container, NULL);
320:     lv_label_set_style(button_col_one, &toggle_tabbtn_pressed);
321:     lv_obj_set_width(button_col_one, CONTROLLER_CONTAINER_WIDTH / 4);
322:     lv_obj_set_height(button_col_one, 20);
323:     lv_label_set_align(button_col_one, LV_LABEL_ALIGN_LEFT);
324:
325:
326: //column two second part that contains function or values
327:     button_col_two = lv_label_create(container, NULL);
328:     lv_label_set_style(button_col_two, &toggle_tabbtn_pressed);
329:     lv_obj_set_width(button_col_two, CONTROLLER_CONTAINER_WIDTH / 4);
330:     lv_obj_set_height(button_col_two, 20);
331:     lv_label_set_align(button_col_two, LV_LABEL_ALIGN_LEFT);
332:
333:
334: //set positions relative to container
335:     lv_obj_align(button_names_one, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
336:     lv_obj_align(button_col_one, container, LV_ALIGN_IN_TOP_MID, -80, 10);
337:
338:     lv_obj_align(button_names_two, container, LV_ALIGN_IN_TOP_MID, 60, 10);
339:     lv_obj_align(button_col_two, container, LV_ALIGN_IN_TOP_RIGHT, -70, 10);
```

```
340:
341: }
342:
343:
344: ControllerTab::~ControllerTab()
345: {
346:     lv_obj_del(button_names_one);
347:     lv_obj_del(button_names_two);
348:     lv_obj_del(button_col_one);
349:     lv_obj_del(button_col_two);
350:
351:     lv_obj_del(container);
352: }
353:
354:
355:
356: /**
357:  * updates the data for the controller tab for either the value of each button
358:  * or the function each button performs by looking at an std::unordered_map contained in
359:  * controller.hpp
360:  */
361: void ControllerTab::update(pros::controller_id_e_t controller, bool showing_values)
362: {
363:     std::string functions_col1 = "";
364:     std::string functions_col2 = "";
365:
366:     std::string values_col1 = "";
367:     std::string values_col2 = "";
368:
369:     if ( controller == pros::E_CONTROLLER_MASTER )
370:     {
371:         //text for functions
372:         functions_col1 = (
373:             Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_LEFT_X) + "\n"
374:             + Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_LEFT_Y) + "\n"
375:             + Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_RIGHT_X) + "\n"
376:             + Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_RIGHT_Y) + "\n"
377:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_L1) + "\n"
378:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_L2) + "\n"
379:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_R2) + "\n"
380:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_R1)
381:         );
382:
383:         functions_col2 = (
384:             Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_UP) + "\n"
385:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_DOWN) + "\n"
386:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_LEFT) + "\n"
387:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_RIGHT) + "\n"
388:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_X) + "\n"
389:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_B) + "\n"
390:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_Y) + "\n"
391:             + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_A)
392:         );
393:
394:         values_col1 = (
395:             std::to_string(Controller::master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_X)) + "\n"
396:             + std::to_string(Controller::master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y)) + "\n"
397:             + std::to_string(Controller::master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_X)) + "\n"
398:             + std::to_string(Controller::master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y)) + "\n"
399:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_L1)) + "\n"
400:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_L2)) + "\n"
401:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_R2)) + "\n"
402:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_R1))
403:         );
404:
405:         values_col2 = (
406:             std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_UP)) + "\n"
407:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_DOWN)) + "\n"
408:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_LEFT)) + "\n"
409:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_RIGHT)) + "\n"
410:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_X)) + "\n"
411:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_B)) + "\n"
412:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_Y)) + "\n"
413:             + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_A))
414:         );
415:
416:
417:     }
418:
419:     else if ( controller == pros::E_CONTROLLER_PARTNER )
420:     {
421:         //text for functions
422:         functions_col1 = (
423:             Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_LEFT_X) + "\n"
424:             + Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_LEFT_Y) + "\n"
425:             + Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_RIGHT_X) + "\n"
426:             + Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_RIGHT_Y) + "\n"
427:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_L1) + "\n"
428:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_L2) + "\n"
429:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_R2) + "\n"
430:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_R1)
431:         );
432:
433:         functions_col2 = (
434:             Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_UP) + "\n"
435:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_DOWN) + "\n"
436:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_LEFT) + "\n"
437:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_RIGHT) + "\n"
438:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_X) + "\n"
439:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_B) + "\n"
440:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_Y) + "\n"
441:             + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_A)
442:         );
443:
444:
445:         values_col1 = (
446:             std::to_string(Controller::partner.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_X)) + "\n"
447:             + std::to_string(Controller::partner.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y)) + "\n"
448:             + std::to_string(Controller::partner.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_X)) + "\n"
449:             + std::to_string(Controller::partner.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y)) + "\n"
450:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_L1)) + "\n"
451:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_L2)) + "\n"
452:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_R2)) + "\n"
```

```
453:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_R1))
454:         );
455:
456:         values_col2 = (
457:             std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_UP)) + "\n"
458:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_DOWN)) + "\n"
459:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_LEFT)) + "\n"
460:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_RIGHT)) + "\n"
461:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_X)) + "\n"
462:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_B)) + "\n"
463:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_Y)) + "\n"
464:             + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_A))
465:         );
466:     }
467:
468:     if ( showing_values )
469:     {
470:         lv_label_set_text(button_col_one, values_col1.c_str());
471:         lv_label_set_text(button_col_two, values_col2.c_str());
472:     }
473:
474:     else
475:     {
476:         lv_label_set_text(button_col_one, functions_col1.c_str());
477:         lv_label_set_text(button_col_two, functions_col2.c_str());
478:     }
479:
480: }
481:
482:
483:
484:
485:
486: ControllerDebug::ControllerDebug()
487: {
488: //reset statics
489:     cont = true;
490:     showing_values = 0;
491:
492:
493: //init screen
494:     controller_debug_screen = lv_obj_create(NULL, NULL);
495:     lv_obj_set_style(controller_debug_screen, &gray);
496:
497: //init title label
498:     title_label = lv_label_create(controller_debug_screen, NULL);
499:     lv_label_set_style(title_label, &heading_text);
500:     lv_obj_set_width(title_label, CONTROLLER_CONTAINER_WIDTH);
501:     lv_obj_set_height(title_label, 20);
502:     lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
503:     lv_label_set_text(title_label, "Controller - Debug");
504:
505: //init tabview
506:     tabview = lv_tabview_create(controller_debug_screen, NULL);
507:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BG, &gray);
508:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_REL, &toggle_tabbtn_released);
509:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_PR, &toggle_tabbtn_pressed);
510:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_INDIC, &sw_indic);
511:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_TGL_REL, &toggle_tabbtn_pressed);
512:     //lv_tabview_set_tab_load_action(tabview,  tab_load_action);
513:     lv_obj_set_width(tabview, CONTROLLER_CONTAINER_WIDTH);
514:     lv_obj_set_height(tabview, 200);
515:
516: //init tabs
517:     general_tab = lv_tabview_add_tab(tabview, "General");
518:     master_tab = lv_tabview_add_tab(tabview, "Master Controller");
519:     partner_tab = lv_tabview_add_tab(tabview, "Partner Controller");
520:
521:
522: //init back button
523:     //button
524:     btn_back = lv_btn_create(controller_debug_screen, NULL);
525:     lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
526:     lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
527:     lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
528:     lv_obj_set_width(btn_back, 75);
529:     lv_obj_set_height(btn_back, 25);
530:
531:     //label
532:     btn_back_label = lv_label_create(btn_back, NULL);
533:     lv_obj_set_style(btn_back_label, &heading_text);
534:     lv_label_set_text(btn_back_label, "Back");
535:
536:
537: //init button to switch between showing values and functions
538:     //button
539:     btn_show_values = lv_btn_create(controller_debug_screen, NULL);
540:     lv_btn_set_style(btn_show_values, LV_BTN_STYLE_REL, &toggle_btn_released);
541:     lv_btn_set_style(btn_show_values, LV_BTN_STYLE_PR, &toggle_btn_pressed);
542:     lv_btn_set_action(btn_show_values, LV_BTN_ACTION_CLICK, btn_show_values_action);
543:     lv_obj_set_width(btn_show_values, 150);
544:     lv_obj_set_height(btn_show_values, 25);
545:
546:     //label
547:     btn_show_values_label = lv_label_create(btn_show_values, NULL);
548:     lv_obj_set_style(btn_show_values_label, &heading_text);
549:     lv_label_set_text(btn_show_values_label, "Show Values");
550:
551:     lv_obj_set_hidden(btn_show_values, true);  //set hidden because button is
552:                                 //not needed on the default tab
553:
554:
555: //init tabs from other classes
556:     GeneralControllerDebugInit(general_tab);
557:
558:
559: //set positions
560:     lv_obj_set_pos(btn_back, 30, 210);
561:     lv_obj_align(btn_show_values, controller_debug_screen, LV_ALIGN_IN_BOTTOM_MID, 0, -5);
562:
563:     lv_obj_set_pos(title_label, 180, 5);
564:
565:     lv_obj_set_pos(tabview, 20, 25);
```

```
566: }
567:
568:
569:
570:
571: ControllerDebug::~ControllerDebug()
572: {
573:     lv_obj_del(controller_debug_screen);
574: }
575:
576:
577:
578:
579: /**
580:  * switches between showing values or function by setting variable to the
581:  * opposite of itself and then it updates the text label based on the new value
582:  */
583:
584: lv_res_t ControllerDebug::btn_show_values_action(lv_obj_t *btn)
585: {
586:     showing_values = !showing_values;
587:     if ( showing_values )
588:     {
589:         lv_label_set_text(btn_show_values_label, "Show Functions");
590:     }
591:     else
592:     {
593:         lv_label_set_text(btn_show_values_label, "Show Values");
594:     }
595:
596:     return LV_RES_OK;
597: }
598:
599:
600:
601:
602: /**
603:  * callback funciton that exits main loop when button is pressed
604:  */
605: lv_res_t ControllerDebug::btn_back_action(lv_obj_t *btn)
606: {
607:     cont = false;
608:     return LV_RES_OK;
609: }
610:
611:
612:
613:
614: /**
615:  * main loop that updates controller information
616:  */
617: void ControllerDebug::debug()
618: {
619:     //used to check if user wants to continue cycling through
620:     //tabs. Will be set to zero and loop will break if user hits
621:     //the back button
622:     cont = true;
623:
624:     lv_tabview_set_tab_act(tabview, 0, NULL);
625:     lv_scr_load(controller_debug_screen);
626:
627:     //init tabs from other classes
628:     ControllerTab controller_tab(master_tab);
629:     ControllerTab controller_tab2(partner_tab);
630:
631:     while ( cont )
632:     {
633:         switch ( lv_tabview_get_tab_act(tabview) ) //switches to tab user wants to go to
634:         {
635:             case 0:
636:                 lv_obj_set_hidden(btn_show_values, true);
637:                 update_general_info();
638:                 break;
639:             case 1:
640:                 lv_obj_set_hidden(btn_show_values, false);
641:                 controller_tab.update(pros::E_CONTROLLER_MASTER, showing_values);
642:                 break;
643:             case 2:
644:                 lv_obj_set_hidden(btn_show_values, false);
645:                 controller_tab2.update(pros::E_CONTROLLER_PARTNER, showing_values);
646:                 break;
647:         }
648:         pros::delay(200);
649:     }
650:
651: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/Debug/FieldControlDebug.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/16/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains class with methods that allow the user to see info about the state
8:   * od the field control
9:   */
10:
11: #ifndef __FIELDCONTROLDEBUG_HPP__
12: #define __FIELDCONTROLDEBUG_HPP__
13:
14:
15: #include "../../../../include/main.h"
16:
17: #include "../Styles.hpp"
18:
19:
20: /**
21:  * @see: ../Styles.hpp
22:  *
23:  * contains methods that show user data about the field control
24:  */
25: class FieldControlDebug : private Styles
26: {
27:     private:
28:         lv_obj_t *field_ctrl_screen;
29:         lv_obj_t *title_label;
30:
31:         lv_obj_t *labels_label;
32:         lv_obj_t *info_label;
33:
34:         //back button
35:         lv_obj_t *btn_back;
36:         lv_obj_t *btn_back_label;
37:
38:         /**
39:          * @param: lv_obj_t* btn -> button that called the funtion
40:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
41:          *
42:          * button callback function used to set cont to false meaning the
43:          * user wants to go to the title screen
44:          */
45:         static lv_res_t btn_back_action(lv_obj_t *btn);
46:
47:     public:
48:         static bool cont;
49:
50:         FieldControlDebug();
51:         ~FieldControlDebug();
52:
53:
54:         /**
55:          * @return: None
56:          * TODO: use ternary operator to condense and make more readable
57:          *
58:          * allows user to see information about the state of field control
59:          */
60:         void debug();
61:
62: };
63:
64:
65:
66:
67: #endif
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/FieldControlDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: FieldControlDebug.hpp
 *
 * contains implementation for class with field control data
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "FieldControlDebug.hpp"


bool FieldControlDebug::cont = true;

FieldControlDebug::FieldControlDebug()
{
    cont = true;

//screen
    field_ctrl_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(field_ctrl_screen, &gray);

//init back button
    //button
    btn_back = lv_btn_create(field_ctrl_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init title label
    title_label = lv_label_create(field_ctrl_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 440);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Field Control - Debug");

//init headers label
    labels_label = lv_label_create(field_ctrl_screen, NULL);
    lv_label_set_style(labels_label, &subheading_text);
    lv_obj_set_width(labels_label, 220);
    lv_obj_set_height(labels_label, 200);
    lv_label_set_align(labels_label, LV_LABEL_ALIGN_LEFT);

    std::string labels_label_text = (
          "connected to competition switch\n"
          "disabled\n"
          "game state"
    );

    lv_label_set_text(labels_label, labels_label_text.c_str());

//init values label
    info_label = lv_label_create(field_ctrl_screen, NULL);
    lv_label_set_style(info_label, &subheading_text);
    lv_obj_set_width(info_label, 220);
    lv_obj_set_height(info_label, 200);
    lv_label_set_align(info_label, LV_LABEL_ALIGN_LEFT);

    std::string info_label_text = (
        "no\n"
        "no\n"
        "no"
    );

    lv_label_set_text(info_label, info_label_text.c_str());

//set positions
    lv_obj_set_pos(btn_back, 30, 210);

    lv_obj_align(title_label, field_ctrl_screen, LV_ALIGN_IN_TOP_MID, 0, 10);

    lv_obj_set_pos(labels_label, 20, 40);
    lv_obj_set_pos(info_label, 360, 40);


}

FieldControlDebug::~FieldControlDebug()
{
    lv_obj_del(field_ctrl_screen);
}


/**
 * sets cont to false to break main loop so main function returns
 */
lv_res_t FieldControlDebug::btn_back_action(lv_obj_t *btn)
{
    cont = false;
    return LV_RES_OK;
}


/**
 * main loop that updates the field control information
 */
void FieldControlDebug::debug()
{
    cont = true;
```

```
114:
115:    lv_scr_load(field_ctrl_screen);
116:
117:    while ( cont )
118:    {
119:       std::string info_label_text = "";
120:
121:       if ( pros::competition::is_connected() )
122:       {
123:          info_label_text = info_label_text + "yes\n";
124:       }
125:       else
126:       {
127:          info_label_text = info_label_text + "no\n";
128:       }
129:
130:       if ( pros::competition::is_disabled() )
131:       {
132:          info_label_text = info_label_text + "yes\n";
133:       }
134:
135:       else
136:       {
137:          info_label_text = info_label_text + "no\n";
138:       }
139:
140:       if ( pros::competition::is_autonomous() )
141:       {
142:          info_label_text = info_label_text + "autonomous\n";
143:       }
144:
145:       else
146:       {
147:          info_label_text = info_label_text + "driver control\n";
148:       }
149:
150:       lv_label_set_text(info_label, info_label_text.c_str());
151:
152:
153:       pros::delay(100);
154:    }
155: }
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/InternalMotorDebug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * contains class that for debugging the internal Motor PID constants
 * from the lcd without having to recompile and upload code
 */

#ifndef __INTERNALMOTORDEBUG_HPP__
#define __INTERNALMOTORDEBUG_HPP__

#include "main.h"

#include "../Styles.hpp"
#include "../../../Configuration.hpp"
#include "../../motors/Motor.hpp"


/**
 * @see: ../../motors/Motor.hpp
 *
 * allows user to tune pid constants by logging data and running unit tests
 * with given parameters
 */
class InternalMotorDebug : private Styles
{
    private:
        static bool cont;
        static bool run;

        Motor motor;

        lv_obj_t *main_screen;
        lv_obj_t *title_label;

    //parameter labels side one
        lv_obj_t *kp_label;
        lv_obj_t *kp_text_area;

        lv_obj_t *ki_label;
        lv_obj_t *ki_text_area;

        lv_obj_t *kd_label;
        lv_obj_t *kd_text_area;

        lv_obj_t *I_max_label;
        lv_obj_t *I_max_text_area;

        lv_obj_t *slew_label;
        lv_obj_t *slew_text_area;

        lv_obj_t *setpoint_label;
        lv_obj_t *setpoint_text_area;

        lv_obj_t *duration_label;
        lv_obj_t *duration_text_area;

    //parameter labels side two
        //port
        lv_obj_t *port_label;
        lv_obj_t *port_text_area;

        //gearset
        static pros::motor_gearset_e_t current_gearset;
        lv_obj_t *gearset_label;
        lv_obj_t *ddlist_gearset;

        /**
         * @param: lv_obj_t* ddlist -> the dropdown list object for the callback function
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * sets the brake mode for the motor set which will be updated in the main loop
         */
        static lv_res_t ddlist_gearset_action(lv_obj_t *ddlist);

        //brake mode
        static pros::motor_brake_mode_e_t current_brake_mode;
        lv_obj_t *brakemode_label;
        lv_obj_t *ddlist_brake_mode;

        /**
         * @param: lv_obj_t* ddlist -> the dropdown list object for the callback function
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * sets the brake mode for the motor set which will be updated in the main loop
         */
        static lv_res_t ddlist_brake_mode_action(lv_obj_t *ddlist);

    //information label
        lv_obj_t *information_label;

    //keyboard

        lv_obj_t *keyboard;


    //back button
        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set cont to false meaning the
         * user wants to go to the title screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);


    //run unit test button
```

```
114:        lv_obj_t *btn_run;
115:        lv_obj_t *btn_run_label;
116:
117:        /**
118:         * @param: lv_obj_t* btn -> button that called the funtion
119:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
120:         *
121:         * button to run unit_test and log data with the given parameters
122:         */
123:        static lv_res_t btn_run_action(lv_obj_t *btn);
124:
125:    //actual unit test function
126:        /**
127:         * @return: int -> 1 if motor was successfully tested, 0 otherwise
128:         *
129:         * reads values from text areas and performs a unit test with the
130:         * given parameters
131:         */
132:        int run_unit_test( );
133:
134:
135:    public:
136:        InternalMotorDebug();
137:        ~InternalMotorDebug();
138:
139:        /**
140:         * @return: None
141:         *
142:         * start debugger
143:         */
144:        void debug();
145:
146:  };
147:
148:
149:
150:  #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/Debug/InternalMotorDebug.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 2/16/2020
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see: InternalMotorDebug.hpp
8:   *
9:   * contains class implementation for tuning the motors internal velocity PID
10:  * controller
11:  */
12:
13:  #include <stdexcept>
14:  #include <string>
15:
16:  #include "main.h"
17:
18:  #include "InternalMotorDebug.hpp"
19:  #include "../Styles.hpp"
20:  #include "../../Configuration.hpp"
21:  #include "../../motors/Motor.hpp"
22:  #include "../../motors/MotorThread.hpp"
23:  #include "../../serial/Logger.hpp"
24:
25:
26:
27:  bool InternalMotorDebug::cont = true;
28:  bool InternalMotorDebug::run = false;
29:  pros::motor_gearset_e_t InternalMotorDebug::current_gearset = pros::E_MOTOR_GEARSET_18;
30:  pros::motor_brake_mode_e_t InternalMotorDebug::current_brake_mode = pros::E_MOTOR_BRAKE_COAST;
31:
32:
33:  InternalMotorDebug::InternalMotorDebug() :
34:      motor(1, pros::E_MOTOR_GEARSET_18, false)
35:  {
36:      MotorThread* motor_thread = MotorThread::get_instance();
37:      motor_thread->register_motor(motor);
38:
39:
40:      cont = true;
41:      run = false;
42:
43:  //screen
44:      main_screen = lv_obj_create(NULL, NULL);
45:      lv_obj_set_style(main_screen, &gray);
46:
47:  //init title label
48:      title_label = lv_label_create(main_screen, NULL);
49:      lv_label_set_style(title_label, &heading_text);
50:      lv_obj_set_width(title_label, 440);
51:      lv_obj_set_height(title_label, 20);
52:      lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
53:      lv_label_set_text(title_label, "Velocity PID Controller - Debugger");
54:
55:
56:      //init parameters side two
57:      //port
58:      port_label = lv_label_create(main_screen, NULL);
59:      lv_label_set_style(port_label, &heading_text);
60:      lv_obj_set_width(port_label, 100);
61:      lv_obj_set_height(port_label, 40);
62:      lv_label_set_align(port_label, LV_LABEL_ALIGN_LEFT);
63:      lv_label_set_text(port_label, "Port");
64:
65:      port_text_area = lv_ta_create(main_screen, NULL);
66:      lv_obj_set_style(port_text_area, &subheading_text);
67:      lv_ta_set_accepted_chars(port_text_area, "0123456789");
68:      lv_obj_set_size(port_text_area, 80, 20);
69:      lv_ta_set_text(port_text_area, "1");
70:      lv_ta_set_one_line(port_text_area, true);
71:
72:      //gearset
73:      current_gearset = pros::E_MOTOR_GEARSET_18;
74:
75:      gearset_label = lv_label_create(main_screen, NULL);
76:      lv_label_set_style(gearset_label, &heading_text);
77:      lv_obj_set_width(gearset_label, 100);
78:      lv_obj_set_height(gearset_label, 40);
79:      lv_label_set_align(gearset_label, LV_LABEL_ALIGN_LEFT);
80:      lv_label_set_text(gearset_label, "Gearset");
81:
82:      ddlist_gearset = lv_ddlist_create(main_screen, NULL);
83:      lv_ddlist_set_options(ddlist_gearset, "100\n200\n600");
84:      lv_obj_set_style(ddlist_gearset, &subheading_text);
85:      lv_obj_set_width(ddlist_gearset, 125);
86:      lv_obj_set_height(ddlist_gearset, 60);
87:      lv_ddlist_set_action(ddlist_gearset, ddlist_gearset_action);
88:      lv_ddlist_set_selected(ddlist_gearset, 1);
89:
90:      //brakemode
91:      current_brake_mode = pros::E_MOTOR_BRAKE_COAST;
92:
93:      brakemode_label = lv_label_create(main_screen, NULL);
94:      lv_label_set_style(brakemode_label, &heading_text);
95:      lv_obj_set_width(brakemode_label, 100);
96:      lv_obj_set_height(brakemode_label, 40);
97:      lv_label_set_align(brakemode_label, LV_LABEL_ALIGN_LEFT);
98:      lv_label_set_text(brakemode_label, "Brakemode");
99:
100:     ddlist_brake_mode = lv_ddlist_create(main_screen, NULL);
101:     lv_ddlist_set_options(ddlist_brake_mode, "Coast\n"
102:                                             "Brake\n"
103:                                             "PID Hold");
104:     lv_obj_set_style(ddlist_brake_mode, &subheading_text);
105:     lv_obj_set_width(ddlist_brake_mode, 125);
106:     lv_obj_set_height(ddlist_brake_mode, 20);
107:     lv_ddlist_set_action(ddlist_brake_mode, ddlist_brake_mode_action);
108:
109:
110: //init parameters side one
111: //kP
112:     kp_label = lv_label_create(main_screen, NULL);
113:     lv_label_set_style(kp_label, &heading_text);
```

```
114:     lv_obj_set_width(kp_label, 440);
115:     lv_obj_set_height(kp_label, 20);
116:     lv_label_set_align(kp_label, LV_LABEL_ALIGN_LEFT);
117:     lv_label_set_text(kp_label, "kP");
118:
119:     kp_text_area = lv_ta_create(main_screen, NULL);
120:     lv_obj_set_style(kp_text_area, &subheading_text);
121:     lv_ta_set_accepted_chars(kp_text_area, ".0123456789");
122:     lv_obj_set_size(kp_text_area, 80, 15);
123:     lv_ta_set_text(kp_text_area, std::to_string(motor.get_pid().kP).c_str());
124:     lv_ta_set_one_line(kp_text_area, true);
125:
126:   //kI
127:     ki_label = lv_label_create(main_screen, NULL);
128:     lv_label_set_style(ki_label, &heading_text);
129:     lv_obj_set_width(ki_label, 100);
130:     lv_obj_set_height(ki_label, 20);
131:     lv_label_set_align(ki_label, LV_LABEL_ALIGN_LEFT);
132:     lv_label_set_text(ki_label, "kI");
133:
134:     ki_text_area = lv_ta_create(main_screen, NULL);
135:     lv_obj_set_style(ki_text_area, &subheading_text);
136:     lv_ta_set_accepted_chars(ki_text_area, ".0123456789");
137:     lv_obj_set_size(ki_text_area, 80, 15);
138:     lv_ta_set_text(ki_text_area, std::to_string(motor.get_pid().kI).c_str());
139:     lv_ta_set_one_line(ki_text_area, true);
140:
141:   //kD
142:     kd_label = lv_label_create(main_screen, NULL);
143:     lv_label_set_style(kd_label, &heading_text);
144:     lv_obj_set_width(kd_label, 100);
145:     lv_obj_set_height(kd_label, 20);
146:     lv_label_set_align(kd_label, LV_LABEL_ALIGN_LEFT);
147:     lv_label_set_text(kd_label, "kD");
148:
149:     kd_text_area = lv_ta_create(main_screen, NULL);
150:     lv_obj_set_style(kd_text_area, &subheading_text);
151:     lv_ta_set_accepted_chars(kd_text_area, ".0123456789");
152:     lv_obj_set_size(kd_text_area, 80, 15);
153:     lv_ta_set_text(kd_text_area, std::to_string(motor.get_pid().kD).c_str());
154:     lv_ta_set_one_line(kd_text_area, true);
155:
156:   //I max
157:     I_max_label = lv_label_create(main_screen, NULL);
158:     lv_label_set_style(I_max_label, &heading_text);
159:     lv_obj_set_width(I_max_label, 100);
160:     lv_obj_set_height(I_max_label, 20);
161:     lv_label_set_align(I_max_label, LV_LABEL_ALIGN_LEFT);
162:     lv_label_set_text(I_max_label, "kI Max");
163:
164:     I_max_text_area = lv_ta_create(main_screen, NULL);
165:     lv_obj_set_style(I_max_text_area, &subheading_text);
166:     lv_ta_set_accepted_chars(I_max_text_area, ".0123456789");
167:     lv_obj_set_size(I_max_text_area, 80, 15);
168:     lv_ta_set_text(I_max_text_area, std::to_string(motor.get_pid().I_max).c_str());
169:     lv_ta_set_one_line(I_max_text_area, true);
170:
171:   //slew rate
172:     slew_label = lv_label_create(main_screen, NULL);
173:     lv_label_set_style(slew_label, &heading_text);
174:     lv_obj_set_width(slew_label, 100);
175:     lv_obj_set_height(slew_label, 40);
176:     lv_label_set_align(slew_label, LV_LABEL_ALIGN_LEFT);
177:     lv_label_set_text(slew_label, "Slew Rate");
178:
179:     slew_text_area = lv_ta_create(main_screen, NULL);
180:     lv_obj_set_style(slew_text_area, &subheading_text);
181:     lv_ta_set_accepted_chars(slew_text_area, "0123456789");
182:     lv_obj_set_size(slew_text_area, 80, 15);
183:     lv_ta_set_text(slew_text_area, std::to_string(motor.get_slew_rate()).c_str());
184:     lv_ta_set_one_line(slew_text_area, true);
185:
186:   //setpoint_label
187:     setpoint_label = lv_label_create(main_screen, NULL);
188:     lv_label_set_style(setpoint_label, &heading_text);
189:     lv_obj_set_width(setpoint_label, 100);
190:     lv_obj_set_height(setpoint_label, 40);
191:     lv_label_set_align(setpoint_label, LV_LABEL_ALIGN_LEFT);
192:     lv_label_set_text(setpoint_label, "Setpoint");
193:
194:     setpoint_text_area = lv_ta_create(main_screen, NULL);
195:     lv_obj_set_style(setpoint_text_area, &subheading_text);
196:     lv_ta_set_accepted_chars(setpoint_text_area, "0123456789");
197:     lv_obj_set_size(setpoint_text_area, 80, 15);
198:     lv_ta_set_text(setpoint_text_area, "200");
199:     lv_ta_set_one_line(setpoint_text_area, true);
200:
201:   //duration
202:     duration_label = lv_label_create(main_screen, NULL);
203:     lv_label_set_style(duration_label, &heading_text);
204:     lv_obj_set_width(duration_label, 100);
205:     lv_obj_set_height(duration_label, 40);
206:     lv_label_set_align(duration_label, LV_LABEL_ALIGN_LEFT);
207:     lv_label_set_text(duration_label, "Duration");
208:
209:     duration_text_area = lv_ta_create(main_screen, NULL);
210:     lv_obj_set_style(duration_text_area, &subheading_text);
211:     lv_ta_set_accepted_chars(duration_text_area, "0123456789");
212:     lv_obj_set_size(duration_text_area, 80, 15);
213:     lv_ta_set_text(duration_text_area, "10000");
214:     lv_ta_set_one_line(duration_text_area, true);
215:
216:
217:
218:   //information label
219:     information_label = lv_label_create(main_screen, NULL);
220:     lv_obj_set_style(information_label, &subheading_text);
221:     lv_label_set_text(information_label, "Info");
222:
223:
224:   //init back button
225:     //button
226:     btn_back = lv_btn_create(main_screen, NULL);
```

```
227:     lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
228:     lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
229:     lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
230:     lv_obj_set_width(btn_back, 75);
231:     lv_obj_set_height(btn_back, 25);
232:
233:     //label
234:     btn_back_label = lv_label_create(btn_back, NULL);
235:     lv_obj_set_style(btn_back_label, &heading_text);
236:     lv_label_set_text(btn_back_label, "Back");
237:
238:
239:   //init run button
240:     //button
241:     btn_run = lv_btn_create(main_screen, NULL);
242:     lv_btn_set_style(btn_run, LV_BTN_STYLE_REL, &toggle_btn_released);
243:     lv_btn_set_style(btn_run, LV_BTN_STYLE_PR, &toggle_btn_pressed);
244:     lv_btn_set_action(btn_run, LV_BTN_ACTION_CLICK, btn_run_action);
245:     lv_obj_set_width(btn_run, 150);
246:     lv_obj_set_height(btn_run, 25);
247:
248:     //label
249:     btn_run_label = lv_label_create(btn_run, NULL);
250:     lv_obj_set_style(btn_run_label, &heading_text);
251:     lv_label_set_text(btn_run_label, "Run Unit Test");
252:
253:
254:   //set up keyboard
255:     keyboard = lv_kb_create(main_screen, NULL);
256:     // lv_kb_set_ta(keyboard, kp_text_area);
257:     // lv_kb_set_ta(keyboard, ki_text_area);
258:     // lv_kb_set_ta(keyboard, kd_text_area);
259:     // lv_kb_set_ta(keyboard, I_max_text_area);
260:     // lv_kb_set_ta(keyboard, slew_text_area);
261:
262:     //lv_ta_set_action(port_text_area, LV_EVENT_PRESSED);
263:
264:   //set positions
265:   //title
266:     lv_obj_set_pos(title_label, 100, 5);
267:
268:   //bottom buttons
269:     lv_obj_set_pos(btn_back, 30, 210);
270:     lv_obj_set_pos(btn_run, 300, 210);
271:
272:   //parameters side 1
273:     lv_obj_set_pos(kp_label, 20, 30);
274:     lv_obj_set_pos(ki_label, 20, 55);
275:     lv_obj_set_pos(kd_label, 20, 80);
276:     lv_obj_set_pos(I_max_label, 20, 105);
277:     lv_obj_set_pos(slew_label, 20, 130);
278:     lv_obj_set_pos(setpoint_label, 20, 155);
279:     lv_obj_set_pos(duration_label, 20, 180);
280:
281:     lv_obj_set_pos(kp_text_area, 130, 23);
282:     lv_obj_set_pos(ki_text_area, 130, 48);
283:     lv_obj_set_pos(kd_text_area, 130, 73);
284:     lv_obj_set_pos(I_max_text_area, 130, 98);
285:     lv_obj_set_pos(slew_text_area, 130, 123);
286:     lv_obj_set_pos(setpoint_text_area, 130, 148);
287:     lv_obj_set_pos(duration_text_area, 130, 173);
288:
289:   //parameters side 2
290:     lv_obj_set_pos(port_label, 240, 40);
291:     lv_obj_set_pos(gearset_label, 240, 75);
292:     lv_obj_set_pos(brakemode_label, 240, 100);
293:
294:     lv_obj_set_pos(port_text_area, 350, 33);
295:     lv_obj_set_pos(ddlist_gearset, 350, 75);
296:     lv_obj_set_pos(ddlist_brake_mode, 350, 100);
297:
298:   //information
299:     lv_obj_set_pos(information_label, 240, 140);
300:
301:
302:
303:   }
304:
305:   InternalMotorDebug::~InternalMotorDebug()
306:   {
307:     MotorThread* motor_thread = MotorThread::get_instance();
308:     motor_thread->unregister_motor(motor);
309:
310:     lv_obj_del(main_screen);
311:   }
312:
313:
314:   /**
315:    * sets cont to false signifying user wants to go back, main loop will exit
316:    */
317:   lv_res_t InternalMotorDebug::btn_back_action(lv_obj_t *btn)
318:   {
319:     cont = false;
320:     return LV_RES_OK;
321:   }
322:
323:   lv_res_t InternalMotorDebug::btn_run_action(lv_obj_t *btn)
324:   {
325:     lv_btn_set_state(btn, LV_BTN_STATE_INA);
326:     run = true;
327:     return LV_RES_OK;
328:   }
329:
330:
331:
332:   /**
333:    * looks at the string of the current drop down list option and compares it to
334:    * a string to see what gearset the user wants
335:    * sets gearset to this value
336:    */
337:   lv_res_t InternalMotorDebug::ddlist_gearset_action(lv_obj_t * ddlist)
338:   {
339:     //checks what the drop down list string is
```

```
340:     char sel_cstr[32];
341:     lv_ddlist_get_selected_str(ddlist, sel_cstr);
342:
343:     std::string sel_str = std::string(sel_cstr);  //convert to std::string so
344:                                                   //that the strings can be
345:                                                   //compared
346:
347:     //sets brake mode for motor
348:     if ( sel_str == "100" )
349:     {
350:        current_gearset = pros::E_MOTOR_GEARSET_36;
351:     }
352:
353:     else if ( sel_str == "600" )
354:     {
355:        current_gearset = pros::E_MOTOR_GEARSET_06;
356:     }
357:
358:     else
359:     {
360:        current_gearset = pros::E_MOTOR_GEARSET_18;
361:     }
362:
363:     return LV_RES_OK; //Return OK because the drop down list was not deleted
364:  }
365:
366:
367:
368:
369:  /**
370:   * looks at the string of the current drop down list option and compares it to
371:   * a string to see what brakemode the user wants
372:   * sets brake mode to this value
373:   */
374:  lv_res_t InternalMotorDebug::ddlist_brake_mode_action(lv_obj_t * ddlist)
375:  {
376:     //checks what the drop down list string is
377:     char sel_cstr[32];
378:     lv_ddlist_get_selected_str(ddlist, sel_cstr);
379:
380:     std::string sel_str = std::string(sel_cstr);  //convert to std::string so
381:                                                   //that the strings can be
382:                                                   //compared
383:
384:     //sets brake mode for motor
385:     if ( sel_str == "PID Hold" )
386:     {
387:        current_brake_mode = pros::E_MOTOR_BRAKE_HOLD;
388:     }
389:
390:     else if ( sel_str == "Brake" )
391:     {
392:        current_brake_mode = pros::E_MOTOR_BRAKE_BRAKE;
393:     }
394:
395:     else
396:     {
397:        current_brake_mode = pros::E_MOTOR_BRAKE_COAST;
398:     }
399:
400:     return LV_RES_OK; //Return OK if the drop down list is not deleted
401:  }
402:
403:
404:
405:
406:  /**
407:   * reads values from text areas and performs data validation, exits on invalid data
408:   * starts unit test and logs data
409:   * updates labels on lcd while waiting for duration to finish
410:   */
411:  int InternalMotorDebug::run_unit_test()
412:  {
413:     Logger logger;
414:
415:     pid pid_constants;
416:
417:     int slew = 0;
418:     int setpoint = 0;
419:     int motor_port = 0;
420:
421:     int duration = 0;
422:
423:     //read info from text areas in exception safe way
424:     try
425:     {
426:        double kP = std::stod(lv_ta_get_text(kp_text_area));
427:        double kI = std::stod(lv_ta_get_text(ki_text_area));
428:        double kD = std::stod(lv_ta_get_text(kd_text_area));
429:        double I_max = std::stod(lv_ta_get_text(I_max_text_area));
430:
431:        pid_constants.kP = kP;
432:        pid_constants.kI = kI;
433:        pid_constants.kD = kD;
434:        pid_constants.I_max = I_max;
435:     }
436:     catch ( const std::invalid_argument& )
437:     {
438:        run = false;
439:
440:        log_entry entry;
441:        entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid pid constants given to internal motor unit test";
442:        entry.stream = "cerr";
443:        logger.add(entry);
444:
445:        return 0;
446:     }
447:
448:     try
449:     {
450:        slew = std::stoi(lv_ta_get_text(slew_text_area));
451:     }
452:     catch ( const std::invalid_argument& )
```

```
453:     {
454:         run = false;
455: 
456:         log_entry entry;
457:         entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid slew rate given to internal motor unit test";
458:         entry.stream = "cerr";
459:         logger.add(entry);
460: 
461:         return 0;
462:     }
463: 
464:     try
465:     {
466:         setpoint = std::stoi(lv_ta_get_text(setpoint_text_area));
467:     }
468:     catch ( const std::invalid_argument& )
469:     {
470:         run = false;
471: 
472:         log_entry entry;
473:         entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid setpoint given to internal motor unit test";
474:         entry.stream = "cerr";
475:         logger.add(entry);
476: 
477:         return 0;
478:     }
479: 
480:     try
481:     {
482:         motor_port = std::stoi(lv_ta_get_text(port_text_area));
483:     }
484:     catch ( const std::invalid_argument& )
485:     {
486:         run = false;
487: 
488:         log_entry entry;
489:         entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid motor port given to internal motor unit test";
490:         entry.stream = "cerr";
491:         logger.add(entry);
492: 
493:         return 0;
494:     }
495: 
496:     try
497:     {
498:         duration = std::stoi(lv_ta_get_text(duration_text_area));
499:     }
500:     catch ( const std::invalid_argument& )
501:     {
502:         run = false;
503: 
504:         log_entry entry;
505:         entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid duration given to internal motor unit test";
506:         entry.stream = "cerr";
507:         logger.add(entry);
508: 
509:         return 0;
510:     }
511: 
512:     //set motor information
513:     motor.set_port(motor_port);
514:     motor.set_gearing(current_gearset);
515:     motor.set_brake_mode(current_brake_mode);
516:     // if ( std::abs(slew) > 0 )
517:     // {
518:     // motor.enable_slew();
519:     // motor.set_slew(30);
520:     // }
521:     // else
522:     // {
523:         motor.disable_slew();
524:     // }
525: 
526:     //motor.disable_velocity_pid();
527:     motor.disable_driver_control();
528:     motor.set_pid( pid_constants );
529:     motor.set_motor_mode(e_custom_velocity_pid);
530:     motor.set_log_level(1);
531:     std::cout << motor.get_pid().kP << "\n";
532: 
533:     MotorThread * motor_thread = MotorThread::get_instance();
534:     motor_thread->start_thread();
535: 
536:     int ut_end_time = pros::millis() + duration;
537:     //motor.move_velocity(setpoint);
538:     motor.move_velocity(200);
539: 
540:     //wait for unit test to finish and update gui in the meantime
541:     while ( pros::millis() < ut_end_time )
542:     {
543:         //update gui
544:         std::string info_str;
545:         info_str = "Voltage: " + std::to_string(motor.get_actual_voltage()) + "\n";
546:         info_str += "Velocity: " + std::to_string(motor.get_actual_velocity()) + "\n";
547:         info_str += "Error: " + std::to_string(setpoint - motor.get_actual_velocity());
548: 
549:         lv_label_set_text(information_label, info_str.c_str());
550:         logger.dump();
551:         pros::delay(50);
552:     }
553: 
554:     motor.set_voltage(0);
555:     pros::delay(2000);
556:     logger.dump();
557:     logger.dump();
558:     logger.dump();
559:     logger.dump();
560:     logger.dump();
561:     logger.dump();
562:     logger.dump();
563:     logger.dump();
564:     logger.dump();
565:     motor.set_log_level(0);
```

```
566:
567:     return 1;
568:   }
569:
570:
571:
572:
573:   /**
574:    * waits for cont to be false which occurs when the user hits the back button
575:    */
576:   void InternalMotorDebug::debug()
577:   {
578:     std::string error_str = "-";
579:     cont = true;
580:     run = false;
581:
582:
583:     lv_scr_load(main_screen);
584:
585:     while ( cont )
586:     {
587:       //update information label
588:       std::string info_str;
589:       info_str = "Voltage: " + std::to_string(motor.get_actual_voltage()) + "\n";
590:       info_str += "Velocity: " + std::to_string(motor.get_actual_velocity()) + "\n";
591:       info_str += "Error: -";
592:       lv_label_set_text(information_label, info_str.c_str());
593:
594:       if ( run )
595:       {
596:         run_unit_test();
597:         run = false;
598:         lv_btn_set_state(btn_run, LV_BTN_STYLE_REL);
599:
600:       }
601:
602:
603:       pros::delay(100);
604:     }
605:
606:   }
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/Debug/MotorsDebug.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/16/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains class that loads tabs to debug motors
8:   */
9:
10:  #ifndef __MOTORDEBUG_HPP__
11:  #define __MOTORDEBUG_HPP__
12:
13:  #include <vector>
14:
15:  #include "../../../../include/main.h"
16:
17:  #include "../Styles.hpp"
18:  #include "../../motors/Motors.hpp"
19:  #include "../../motors/Motor.hpp"
20:
21:  //user defines
22:
23:  //sets size of container
24:  #define MOTORS_CONTAINER_WIDTH 440
25:  #define MOTORS_CONTAINER_HEIGHT 100
26:
27:  //sets percent at which to step velocity at
28:  //10 is reasonable because anything higher gives
29:  //less control and anything lower will make it
30:  //difficult to ramp up or down
31:  #define STEP_PERCENT 10
32:
33:
34:
35:  /**
36:   * @see: ../Styles.hpp
37:   *
38:   * general tab for one or two motors max
39:   * contains methods to show data and set velocity of motors
40:   * on this tab
41:   */
42:  class MotorsDebugTab : virtual Styles
43:  {
44:      private:
45:          lv_obj_t *container;
46:          lv_obj_t *motor1_label;
47:          lv_obj_t *motor2_label;
48:
49:          lv_obj_t *motor1_info;
50:          lv_obj_t *motor2_info;
51:
52:          std::vector<Motor*> motors;
53:          std::vector<std::string> titles;
54:
55:      public:
56:          MotorsDebugTab( std::vector<Motor*> motors_vec, std::vector<std::string> titles_vec, lv_obj_t *parent);
57:          ~MotorsDebugTab();
58:
59:          /**
60:           * @param: int target_velocity -> velocity the motor should be set to
61:           * @param: lv_obj_t* velocity_label -> label that current veolicty will be written to
62:           * @return: None
63:           *
64:           * @see: ../Styles.hpp
65:           * @see: ../../motors/Motors.hpp
66:           *
67:           * updates text for the motors that the class was instatiated with
68:           * also sets the velocity of the motor to int target_velocity
69:           * data shown is current drawn, voltage, reversed or not, temperature, encoder value,
70:           * and tourque
71:           */
72:          void update_label(int target_velocity, lv_obj_t *velocity_label);
73:  };
74:
75:
76:
77:
78:  /**
79:   * @see: class MotorsDebugTab
80:   * @see: ../Styles.hpp
81:   *
82:   * contatins debugger for motors
83:   * gives data for each motor set ie. left chassis, right chassis, intake, etc.
84:   */
85:  class MotorsDebug : virtual Styles
86:  {
87:      private:
88:          //screen
89:          lv_obj_t *motor_debug_screen;
90:
91:          //title label
92:          lv_obj_t *title_label;
93:
94:          //back button
95:          lv_obj_t *btn_back;
96:          lv_obj_t *btn_back_label;
97:
98:          /**
99:           * @param: lv_obj_t* btn -> button that called the funtion
100:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
101:          *
102:          * button callback function used to set cont to false meaning the
103:          * user wants to go to the title screen
104:          */
105:         static lv_res_t btn_back_action(lv_obj_t *btn);
106:
107:
108:         static lv_obj_t *tabview;   //tabview object
109:
110:         lv_obj_t *l_chassis_tab;    //individual tabs
111:         lv_obj_t *r_chassis_tab;    //content will come from base classes
112:         lv_obj_t *main_intake_tab;
113:         lv_obj_t *front_intake_tab;
```

```
114:
115:        static uint16_t tab_loaded;  // 0 = left chassis
116:                                     // 1 = right chassis
117:                                     // 2 = main intake
118:                                     // 3 = front intake
119:
120:        /**
121:         * @param: lv_obj_t* tabview -> tabview object for callback function
122:         * @param: uint16_t -> id of active tab
123:         * @return: lv_res_t -> return LV_RES_OK since object was not deleted
124:         *
125:         * funtion to stop motor movements and set the ability for other threads
126:         * to limit the speed of the motor ie. set it to zero in driver control
127:         * also updates target velocity and the tab loaded
128:         */
129:        static lv_res_t tab_load_action(lv_obj_t *tabview, uint16_t act_id);
130:
131:
132:        //velocity setting buttons
133:        lv_obj_t *velocity_label;
134:
135:        lv_obj_t *btn_pos_increase;
136:        lv_obj_t *btn_neg_increase;
137:        lv_obj_t *btn_stp;
138:
139:        lv_obj_t *btn_pos_increase_label;
140:        lv_obj_t *btn_neg_increase_label;
141:        lv_obj_t *btn_stp_label;
142:
143:
144:        /**
145:         * @param: lv_obj_t* btn -> button that called the funtion
146:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
147:         *
148:         * @see: std::tuple<int, int> get_velocity_step()
149:         *
150:         * button callback function used to decrease the target velocity
151:         */
152:        static lv_res_t btn_pos_increase_action(lv_obj_t *btn);
153:
154:
155:        /**
156:         * @param: lv_obj_t* btn -> button that called the funtion
157:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
158:         *
159:         * @see: std::tuple<int, int> get_velocity_step()
160:         *
161:         * button callback function used to increase the target velocity
162:         */
163:        static lv_res_t btn_neg_increase_action(lv_obj_t *btn);
164:
165:
166:        /**
167:         * @param: lv_obj_t* btn -> button that called the funtion
168:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
169:         *
170:         * @see: std::tuple<int, int> get_velocity_step()
171:         *
172:         * button callback function used to set target velocity to zero
173:         */
174:        static lv_res_t btn_stp_action(lv_obj_t *btn);
175:
176:
177:        /**
178:         * @return: std::tuple<int, int> -> tuple of step, a percentage of max velocity
179:         *          based on STEP_PERCENT, and max velocity of the motor
180:         * TOOD: update max velocity for motors and make more adaptable to changing motors
181:         *
182:         * gets the amount the step should be and the max velocity for the motor
183:         * the max velocity is higher than actual because the motor can go faster
184:         * than the specified RPM
185:         */
186:        static std::tuple<int, int> get_velocity_step();
187:
188:
189:        //static vars to help keep velocity
190:        //need to be static because they will be modified by
191:        //static function
192:        static int target_velocity;
193:
194:
195:        //brake mode option widgets
196:        lv_obj_t *brake_mode_label;
197:        lv_obj_t *ddlist_brake_mode;
198:
199:
200:        /**
201:         * @param: lv_obj_t* ddlist -> the dropdown list object for the callback function
202:         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
203:         *
204:         * sets the brake mode for the motor set which will be updated in the main loop
205:         */
206:        static lv_res_t ddlist_brake_mode_action(lv_obj_t *ddlist);
207:
208:        static pros::motor_brake_mode_e_t current_brake_mode;
209:
210:    public:
211:        MotorsDebug();
212:        ~MotorsDebug();
213:
214:        static bool cont; //checks whether to keep letting user
215:                          //cycle through tabs
216:
217:        /**
218:         * @return: None
219:         *
220:         * allows user to interact with tabs for each motor set that display
221:         * data about those motors
222:         */
223:        void debug();
224:
225:    };
226:
```

```
227:
228: #endif
```

```
  1: /**
  2:  * @file: ./RobotCode/src/lcdCode/Debug/MotorsDebug.cpp
  3:  * @author: Aiden Carney
  4:  * @reviewed_on: 10/16/2019
  5:  * @reviewed_by: Aiden Carney
  6:  *
  7:  * @see: MotorsDebug.hpp
  8:  *
  9:  * contains classes and methods implementation that allow the gui to show
 10:  * the user information about groups of motors seperated into tabs
 11:  */
 12:
 13: #include <cstdint>
 14: #include <cmath>
 15: #include <vector>
 16:
 17: #include "../../../../include/main.h"
 18: #include "../../../../include/api.h"
 19:
 20: #include "../Styles.hpp"
 21: #include "../../motors/Motors.hpp"
 22: #include "MotorsDebug.hpp"
 23:
 24:
 25: //declare static members of all classes
 26: bool MotorsDebug::cont = true;
 27: int MotorsDebug::target_velocity = 0;
 28: lv_obj_t *MotorsDebug::tabview;
 29: pros::motor_brake_mode_e_t MotorsDebug::current_brake_mode = pros::E_MOTOR_BRAKE_COAST;
 30: uint16_t MotorsDebug::tab_loaded = 0;
 31:
 32:
 33:
 34: MotorsDebugTab::MotorsDebugTab( std::vector<Motor*> motors_vec, std::vector<std::string> titles_vec, lv_obj_t *parent)
 35: {
 36:    for( int i = 0; i < motors_vec.size(); i++ )
 37:    {
 38:       motors.push_back(motors_vec.at(i));
 39:       titles.push_back(titles_vec.at(i));
 40:    }
 41: //init container
 42:    container = lv_cont_create(parent, NULL);
 43:    lv_cont_set_fit(container, false, false);
 44:    lv_obj_set_style(container, &gray);
 45:    lv_cont_set_fit(container, false, false);
 46:    lv_obj_set_width(container, MOTORS_CONTAINER_WIDTH);
 47:    lv_obj_set_height(container, MOTORS_CONTAINER_HEIGHT);
 48:
 49: //init motor 1 label
 50:    motor1_label = lv_label_create(container, NULL);
 51:    lv_obj_set_style(motor1_label, &toggle_tabbtn_pressed);
 52:    lv_obj_set_width(motor1_label, (MOTORS_CONTAINER_WIDTH/2));
 53:    lv_obj_set_height(motor1_label, 20);
 54:    lv_label_set_align(motor1_label, LV_LABEL_ALIGN_CENTER);
 55:    lv_label_set_text(motor1_label, titles.at(0).c_str());
 56:
 57: //init motor 1 info label
 58:    motor1_info = lv_label_create(container, NULL);
 59:    lv_obj_set_style(motor1_info, &toggle_tabbtn_pressed);
 60:    lv_obj_set_width(motor1_info, (MOTORS_CONTAINER_WIDTH/2));
 61:    lv_obj_set_height(motor1_info, 20);
 62:    lv_label_set_align(motor1_info, LV_LABEL_ALIGN_LEFT);
 63:    lv_label_set_text(motor1_info, "None\nNone\nNone\nNone\nNone\nNone");
 64:
 65: //init motor 2 label
 66:    motor2_label = lv_label_create(container, NULL);
 67:    lv_obj_set_style(motor2_label, &toggle_tabbtn_pressed);
 68:    lv_obj_set_width(motor2_label, (MOTORS_CONTAINER_WIDTH/2));
 69:    lv_obj_set_height(motor2_label, 20);
 70:    lv_label_set_align(motor2_label, LV_LABEL_ALIGN_CENTER);
 71:    lv_label_set_text(motor2_label, "");
 72:
 73:    motor2_info = lv_label_create(container, NULL);
 74:    lv_obj_set_style(motor2_info, &toggle_tabbtn_pressed);
 75:    lv_obj_set_width(motor2_info, (MOTORS_CONTAINER_WIDTH/2));
 76:    lv_obj_set_height(motor2_info, 20);
 77:    lv_label_set_align(motor2_info, LV_LABEL_ALIGN_LEFT);
 78:    lv_label_set_text(motor2_info, "None");
 79:
 80:
 81:
 82:    if( motors.size() > 1 )
 83:    {
 84:       lv_label_set_text(motor2_label, titles.at(1).c_str());
 85:    }
 86:
 87: //align objects on container
 88:    lv_obj_set_pos(motor1_label, 60, 0);
 89:    lv_obj_set_pos(motor1_info, 10, 15);
 90:    if( motors.size() > 1 )
 91:    {
 92:       lv_obj_set_pos(motor2_label, 315, 0);
 93:       lv_obj_set_pos(motor2_info, 255, 15);
 94:    }
 95: }
 96:
 97: MotorsDebugTab::~MotorsDebugTab()
 98: {
 99:    lv_obj_del(motor1_label);
100:    lv_obj_del(motor1_info);
101:    lv_obj_del(motor2_label);
102:    lv_obj_del(motor2_info);
103:
104:    lv_obj_del(container);
105: }
106:
107:
108:
109: /**
110:  * function to be called in main loop so that data about motors will be updated
111:  * sets velocity, updates data, and updates velocity label
112:  */
113: void MotorsDebugTab::update_label(int target_velocity, lv_obj_t *velocity_label)
```

```
114: {
115:     std::string info1 = "";
116:     std::string info2 = "";
117: 
118:  //set velocity to move at
119:     std::int32_t vel = target_velocity;
120:     motors.at(0)->move_velocity(vel);
121:     if ( motors.size() > 1 )
122:     {
123:         motors.at(1)->move_velocity(vel);
124:     }
125: 
126:  //info for first motor
127:     info1 += "Current Draw: " + std::to_string(motors.at(0)->get_current_draw()) + "\n";
128:     info1 += "Voltage (mV): " + std::to_string(motors.at(0)->get_actual_voltage()) + "\n";
129:     info1 += "State: ";
130:     info1 += motors.at(0)->is_reversed() ? "reversed\n" : "not reversed\n";
131:     info1 += "Temperature: " + std::to_string(motors.at(0)->get_temperature()) + "\n";
132:     info1 += "Encoder Position: " + std::to_string(motors.at(0)->get_encoder_position()) + "\n";
133:     info1 += "Torque (Nm): " + std::to_string(motors.at(0)->get_torque()) + "\n";
134: 
135:  //info for second motor if it exists
136:     if ( motors.size() > 1 )
137:     {
138:         info2 += "Current Draw: " + std::to_string(motors.at(1)->get_current_draw()) + "\n";
139:         info2 += "Voltage (mV): " + std::to_string(motors.at(1)->get_actual_voltage()) + "\n";
140:         info2 += "State: ";
141:         info2 += motors.at(1)->is_reversed() ? "reversed\n" : "not reversed\n";
142:         info2 += "Temperature: " + std::to_string(motors.at(1)->get_temperature()) + "\n";
143:         info2 += "Encoder Position: " + std::to_string(motors.at(1)->get_encoder_position()) + "\n";
144:         info2 += "Torque (Nm): " + std::to_string(motors.at(1)->get_torque()) + "\n";
145:     }
146:  //info for velocity label
147:     std::string velocity;
148:     velocity += titles.at(0) + ": " + std::to_string(motors.at(0)->get_actual_velocity()) + "\n";
149:     if ( motors.size() > 1 )
150:     {
151:         velocity += titles.at(1) + ": " + std::to_string(motors.at(1)->get_actual_velocity());
152:     }
153: 
154:  //set labels
155:     //casts info strings to c strings to make them compatible with lvgl
156:     lv_label_set_text(motor1_info, info1.c_str());
157:     if ( motors.size() > 1 )
158:     {
159:         lv_label_set_text(motor2_info, info2.c_str());
160:     }
161:     lv_label_set_text(velocity_label, velocity.c_str());
162: 
163: }
164: 
165: 
166: 
167: 
168: MotorsDebug::MotorsDebug()
169: {
170:     //set default for statics
171:     cont = true;
172:     target_velocity = 0;
173:     tab_loaded = 0;
174: 
175:  //init screen
176:     motor_debug_screen = lv_obj_create(NULL, NULL);
177:     lv_obj_set_style(motor_debug_screen, &gray);
178: 
179:  //init title label
180:     title_label = lv_label_create(motor_debug_screen, NULL);
181:     lv_label_set_style(title_label, &heading_text);
182:     lv_obj_set_width(title_label, MOTORS_CONTAINER_WIDTH);
183:     lv_obj_set_height(title_label, 20);
184:     lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
185:     lv_label_set_text(title_label, "Motors - Debug");
186: 
187:  //init tabview
188:     tabview = lv_tabview_create(motor_debug_screen, NULL);
189:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BG, &gray);
190:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_REL, &toggle_tabbtn_released);
191:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_PR, &toggle_tabbtn_pressed);
192:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_INDIC, &sw_indic);
193:     lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_TGL_REL, &toggle_tabbtn_pressed);
194:     lv_tabview_set_tab_load_action(tabview, tab_load_action);
195:     lv_obj_set_width(tabview, MOTORS_CONTAINER_WIDTH);
196:     lv_obj_set_height(tabview, 200);
197: 
198:  //init tabs
199:     l_chassis_tab = lv_tabview_add_tab(tabview, "Chassis (L)");
200:     r_chassis_tab = lv_tabview_add_tab(tabview, "Chassis (R)");
201:     main_intake_tab = lv_tabview_add_tab(tabview, "Main Intake");
202:     front_intake_tab = lv_tabview_add_tab(tabview, "Front Intakes");
203: 
204:  //init back button
205:     //button
206:     btn_back = lv_btn_create(motor_debug_screen, NULL);
207:     lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
208:     lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
209:     lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
210:     lv_obj_set_width(btn_back, 75);
211:     lv_obj_set_height(btn_back, 25);
212: 
213:     //label
214:     btn_back_label = lv_label_create(btn_back, NULL);
215:     lv_obj_set_style(btn_back_label, &heading_text);
216:     lv_label_set_text(btn_back_label, "Back");
217: 
218:  //init velocity label
219:     velocity_label = lv_label_create(motor_debug_screen, NULL);
220:     lv_obj_set_style(velocity_label, &subheading_text);
221:     lv_obj_set_width(velocity_label, 100);
222:     lv_obj_set_height(velocity_label, 40);
223:     lv_label_set_align(velocity_label, LV_LABEL_ALIGN_LEFT);
224:     lv_label_set_text(velocity_label, "Velocity: ");
225: 
226:  //init velocity increase button
```

```
227:     //button
228:     btn_pos_increase =  lv_btn_create(motor_debug_screen, NULL);
229:     lv_btn_set_style(btn_pos_increase, LV_BTN_STYLE_REL, &toggle_btn_released);
230:     lv_btn_set_style(btn_pos_increase, LV_BTN_STYLE_PR, &toggle_btn_pressed);
231:     lv_btn_set_action(btn_pos_increase, LV_BTN_ACTION_CLICK, btn_pos_increase_action);
232:     lv_obj_set_width(btn_pos_increase, 40);
233:     lv_obj_set_height(btn_pos_increase, 25);
234:
235:     //label
236:     btn_pos_increase_label = lv_label_create(btn_pos_increase, NULL);
237:     lv_obj_set_style(btn_pos_increase_label, &heading_text);
238:     lv_label_set_text(btn_pos_increase_label, SYMBOL_RIGHT);
239:
240:   //init velocity decrease button
241:     //button
242:     btn_neg_increase =  lv_btn_create(motor_debug_screen, NULL);
243:     lv_btn_set_style(btn_neg_increase, LV_BTN_STYLE_REL, &toggle_btn_released);
244:     lv_btn_set_style(btn_neg_increase, LV_BTN_STYLE_PR, &toggle_btn_pressed);
245:     lv_btn_set_action(btn_neg_increase, LV_BTN_ACTION_CLICK, btn_neg_increase_action);
246:     lv_obj_set_width(btn_neg_increase, 40);
247:     lv_obj_set_height(btn_neg_increase, 25);
248:
249:     //label
250:     btn_neg_increase_label = lv_label_create(btn_neg_increase, NULL);
251:     lv_obj_set_style(btn_neg_increase_label, &heading_text);
252:     lv_label_set_text(btn_neg_increase_label, SYMBOL_LEFT);
253:
254:   //init zero velocity button
255:     //button
256:     btn_stp =  lv_btn_create(motor_debug_screen, NULL);
257:     lv_btn_set_style(btn_stp, LV_BTN_STYLE_REL, &toggle_btn_released);
258:     lv_btn_set_style(btn_stp, LV_BTN_STYLE_PR, &toggle_btn_pressed);
259:     lv_btn_set_action(btn_stp, LV_BTN_ACTION_CLICK, btn_stp_action);
260:     lv_obj_set_width(btn_stp, 40);
261:     lv_obj_set_height(btn_stp, 25);
262:
263:     //label
264:     btn_stp_label = lv_label_create(btn_stp, NULL);
265:     lv_obj_set_style(btn_stp_label, &heading_text);
266:     lv_label_set_text(btn_stp_label, SYMBOL_STOP);
267:
268:   //init brake mode label
269:     brake_mode_label = lv_label_create(motor_debug_screen, NULL);
270:     lv_obj_set_style(brake_mode_label, &heading_text);
271:     lv_obj_set_width(brake_mode_label, 100);
272:     lv_obj_set_height(brake_mode_label, 20);
273:     lv_label_set_align(brake_mode_label, LV_LABEL_ALIGN_CENTER);
274:     lv_label_set_text(brake_mode_label, "Brakemode: ");
275:
276:   //init drop down list
277:     ddlist_brake_mode = lv_ddlist_create(motor_debug_screen, NULL);
278:     lv_ddlist_set_options(ddlist_brake_mode, "Coast\n"
279:                                              "Brake\n"
280:                                              "PID Hold");
281:     lv_obj_set_style(ddlist_brake_mode, &subheading_text);
282:     lv_obj_set_width(ddlist_brake_mode, 125);
283:     lv_obj_set_height(ddlist_brake_mode, 18);
284:     lv_ddlist_set_action(ddlist_brake_mode, ddlist_brake_mode_action);
285:
286:   //set positions
287:     lv_obj_set_pos(btn_back, 30, 210);
288:     lv_obj_set_pos(btn_pos_increase, 270, 210);
289:     lv_obj_set_pos(btn_stp, 220, 210);
290:     lv_obj_set_pos(btn_neg_increase, 170, 210);
291:
292:     lv_obj_set_pos(velocity_label, 330, 177);
293:
294:     lv_obj_set_pos(brake_mode_label, 60, 177);
295:     lv_obj_set_pos(ddlist_brake_mode, 170, 177);
296:
297:     lv_obj_set_pos(title_label, 180, 5);
298:
299:     lv_obj_set_pos(tabview, 20, 25);
300:
301:   }
302:
303:
304:   MotorsDebug::~MotorsDebug()
305:   {
306:     //sets motors to off
307:     Motors::stop_all_motors();
308:
309:     //allow motor to go to zero for driver control if it is not set
310:     //already
311:     Motors::enable_driver_control();
312:
313:     //deletes widgets instantiated by class
314:     lv_obj_del(title_label);
315:
316:     lv_obj_del(btn_back_label);
317:     lv_obj_del(btn_back);
318:
319:     lv_obj_del(l_chassis_tab);
320:     lv_obj_del(r_chassis_tab);
321:     lv_obj_del(main_intake_tab);
322:     lv_obj_del(front_intake_tab);
323:     lv_obj_del(tabview);
324:
325:     lv_obj_del(velocity_label);
326:
327:     lv_obj_del(btn_pos_increase_label);
328:     lv_obj_del(btn_neg_increase_label);
329:     lv_obj_del(btn_stp_label);
330:     lv_obj_del(btn_pos_increase);
331:     lv_obj_del(btn_neg_increase);
332:     lv_obj_del(btn_stp);
333:
334:     lv_obj_del(brake_mode_label);
335:     lv_obj_del(ddlist_brake_mode);
336:
337:
338:     lv_obj_del(motor_debug_screen);
339:   }
```

```
340:
341:
342:  /**
343:   * set cont to false to break main loop
344:   */
345:  lv_res_t MotorsDebug::btn_back_action(lv_obj_t *btn)
346:  {
347:      cont = false;
348:      return LV_RES_OK;
349:  }
350:
351:
352:  /**
353:   * callback function for when a new tab is selected
354:   * used to set motor to default ie. brakemode, velocity
355:   */
356:  lv_res_t MotorsDebug::tab_load_action(lv_obj_t *tabview, uint16_t act_id)
357:  {
358:      tab_loaded = act_id;
359:      target_velocity = 0;
360:
361:      //sets motors to off
362:      Motors::stop_all_motors();
363:
364:      //allow motor to go to zero for driver control if it is not set
365:      //already
366:      Motors::enable_driver_control();
367:
368:      return LV_RES_OK;
369:
370:  }
371:
372:
373:
374:  /**
375:   * looks at the current tab loaded to decide on max velocity because the motor
376:   * can be determined from that
377:   * gets the step percent by looking at what STEP_PERCENT is defined as
378:   */
379:  std::tuple<int, int> MotorsDebug::get_velocity_step()
380:  {
381:      int index = tab_loaded;    // 0 = left chassis - 200RPM
382:                                 // 1 = right chassis - 200RPM
383:                                 // 2 = tilter - 100RPM
384:                                 // 3 = intake - 100RPM
385:                                 // 4 = lift - 100RPM
386:      int max;
387:
388:      if ( index == 0 || index == 1 )
389:      {
390:          max = 250;
391:      }
392:
393:      else if (index == 2 || index == 3 || index == 4)
394:      {
395:          max = 130;
396:      }
397:
398:      else
399:      {
400:          max = 650;
401:      }
402:
403:      int step = static_cast<int>(max / STEP_PERCENT);
404:      return std::make_tuple(step, max);
405:  }
406:
407:
408:  /**
409:   * increses velocity of motor by calling get_velocity_step but limits it to
410:   * the max velocity
411:   */
412:  lv_res_t MotorsDebug::btn_pos_increase_action(lv_obj_t *btn)
413:  {
414:      //increases velocity by user defined percent
415:      int step;
416:      int max;
417:
418:      std::tie(step, max) = get_velocity_step();
419:      if ( target_velocity < max )
420:      {
421:          target_velocity = target_velocity + step;
422:      }
423:
424:      return LV_RES_OK;
425:  }
426:
427:
428:  /**
429:   * decreases velocity of motor by calling get_velocity_step but limits it to
430:   * the max velocity in the negative direction
431:   */
432:  lv_res_t MotorsDebug::btn_neg_increase_action(lv_obj_t *btn)
433:  {
434:      //decreases velocity by user defined percent
435:      int step;
436:      int max;
437:
438:      std::tie(step, max) = get_velocity_step();
439:      if ( target_velocity > 0-max )
440:      {
441:          target_velocity = target_velocity - step;
442:      }
443:
444:      return LV_RES_OK;
445:  }
446:
447:
448:  /**
449:   * sets velocity of motor to zero, used so that user does not have to click
450:   * many times to stop the motor
451:   */
452:  lv_res_t MotorsDebug::btn_stp_action(lv_obj_t *btn)
```

```
453:  {
454:      target_velocity = 0;
455:      return LV_RES_OK;
456:  }
457:
458:
459:  /**
460:   * looks at the string of the current drop down list option and compares it to
461:   * a string to see what brakemode the user wants
462:   * sets brake mode to this value
463:   */
464:  lv_res_t MotorsDebug::ddlist_brake_mode_action(lv_obj_t * ddlist)
465:  {
466:      //checks what the drop down list string is
467:      char sel_cstr[32];
468:      lv_ddlist_get_selected_str(ddlist, sel_cstr);
469:
470:      std::string sel_str = std::string(sel_cstr);  //convert to std::string so
471:                                                    //that the strings can be
472:                                                    //compared
473:
474:      //sets brake mode for motor
475:      if ( sel_str == "PID Hold" )
476:      {
477:          current_brake_mode = pros::E_MOTOR_BRAKE_HOLD;
478:      }
479:
480:      else if ( sel_str == "Brake" )
481:      {
482:          current_brake_mode = pros::E_MOTOR_BRAKE_BRAKE;
483:      }
484:
485:      else
486:      {
487:          current_brake_mode = pros::E_MOTOR_BRAKE_COAST;
488:      }
489:
490:      return LV_RES_OK; //Return OK if the drop down list is not deleted
491:  }
492:
493:
494:
495:
496:  /**
497:   * has a main loop that updates internal data as user cycles through tabs
498:   * to keep data relevent and motors following the function they are supposed to
499:   * loads tabs for each motor set
500:   */
501:  void MotorsDebug::debug()
502:  {
503:      //used to check if user wants to continue cycling through
504:      //tabs. Will be set to zero and loop will break if user hits
505:      //the back button
506:      cont = true;
507:
508:      lv_tabview_set_tab_act(tabview, 0, NULL);
509:      lv_scr_load(motor_debug_screen);
510:
511:      MotorsDebugTab l_chassis_tab_debug( {&Motors::front_left, &Motors::back_left}, {"Front Left", "Back Left"}, l_chassis_tab );
512:      MotorsDebugTab r_chassis_tab_debug( {&Motors::front_right, &Motors::back_right}, {"Front Right", "Back Right"}, r_chassis_tab );
513:      MotorsDebugTab main_intake_tab_debug( {&Motors::upper_indexer, &Motors::lower_indexer}, {"upper_indexer", "lower_indexer"}, main_intake_tab );
514:      MotorsDebugTab front_intake_tab_debug( {&Motors::left_intake, &Motors::right_intake}, {"Left Intake", "Right Intake"}, front_intake_tab );
515:
516:      while ( cont )
517:      {
518:
519:          switch ( tab_loaded ) //switches to tab user wants to go to
520:          {
521:              case 0:
522:                  l_chassis_tab_debug.update_label(target_velocity, velocity_label);
523:                  break;
524:              case 1:
525:                  r_chassis_tab_debug.update_label(target_velocity, velocity_label);
526:                  break;
527:              case 2:
528:                  main_intake_tab_debug.update_label(target_velocity, velocity_label);
529:                  break;
530:              case 3:
531:                  front_intake_tab_debug.update_label(target_velocity, velocity_label);
532:                  break;
533:
534:          }
535:
536:
537:          Motors::set_brake_mode(current_brake_mode);
538:
539:
540:
541:          pros::delay(200);
542:      }
543:
544:      //reallow motor to hit zero velocity for driver controll
545:      Motors::enable_driver_control();
546:  }
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/Debug/SensorsDebug.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   * TODO: condense, there are several classes that could be combined so that their is not so many
7:   *
8:   * contains classes for tabs of the sensors debugger tab
9:   */
10:
11:  #ifndef __SENSORDEBUG_HPP__
12:  #define __SENSORDEBUG_HPP__
13:
14:  #include <string>
15:
16:  #include "../../../../include/main.h"
17:
18:  #include "../Styles.hpp"
19:  #include "../Gimmicks.hpp"
20:  #include "../../motors/Motors.hpp"
21:  #include "../../sensors/Sensors.hpp"
22:
23:
24:  //user defines
25:
26:  //sets size of container
27:  #define SENSORS_CONTAINER_WIDTH 440
28:  #define SENSORS_CONTAINER_HEIGHT 120
29:
30:
31:  //Base classes
32:  //base classes are the tabs that will be loaded by the derived class
33:  //this makes it easy to add new tabs while keeping the amount that has
34:  //to go in one class to a minimum, especially since lvgl is not light
35:
36:
37:
38:  // /**
39:  //  * @see: ../Styles.hpp
40:  //  *
41:  //  * shows tab of IMEs and allows user to tare encoders and see values
42:  //  */
43:  // class IMEsDebugger :
44:  //     virtual Styles
45:  // {
46:  //     private:
47:  //         lv_obj_t *container;
48:  //         lv_obj_t *title;
49:  //         lv_obj_t *info;
50:  //
51:  //         lv_obj_t *btn_tare;
52:  //         lv_obj_t *btn_tare_label;
53:  //
54:  //         /**
55:  //          * @param: lv_obj_t* btn -> button that called the funtion
56:  //          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
57:  //          *
58:  //          * button callback function used to tare all IMEs
59:  //          */
60:  //         static lv_res_t btn_tare_action(lv_obj_t *btn);
61:  //
62:  //
63:  //     protected:
64:  //         /**
65:  //          * @return: None
66:  //          *
67:  //          * updates values for IMEs
68:  //          */
69:  //         void update_imes_info();
70:  //
71:  //     public:
72:  //         IMEsDebugger();
73:  //         virtual ~IMEsDebugger();
74:  //
75:  //         /**
76:  //          * @param: lv_obj_t* parent -> parent of the tab
77:  //          * @return: None
78:  //          *
79:  //          * objects are initially loaded onto a NULL parent to be updated later
80:  //          * this sets it so that the parent of the objects is now the tab
81:  //          */
82:  //         void IMEsDebuggerInit(lv_obj_t *parent);
83:  //
84:  // };
85:
86:
87:  // /**
88:  //  * @see: ../Styles.
89:  //  *
90:  //  * show value for potentiometer
91:  //  */
92:  // class PotentiometerDebugger :
93:  //     virtual Styles
94:  // {
95:  //     private:
96:  //         lv_obj_t *container;
97:  //
98:  //         lv_obj_t *title1;
99:  //         lv_obj_t *title2;
100: //         lv_obj_t *title3;
101: //
102: //         lv_obj_t *info1;
103: //         lv_obj_t *info2;
104: //         lv_obj_t *info3;
105: //
106: //         lv_obj_t *btn_calibrate;
107: //         lv_obj_t *btn_calibrate_label;
108: //
109: //         /**
110: //          * @param: lv_obj_t* btn -> button that called the funtion
111: //          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
112: //          *
113: //          * button callback function used to calibrate sensor
```

```
114:  //         */
115:  //        static lv_res_t btn_calibrate_action(lv_obj_t *btn);
116:  //
117:  //    protected:
118:  //        /**
119:  //         * @return: None
120:  //         *
121:  //         * updates value of potentiometer
122:  //         */
123:  //        void update_pot_info();
124:  //
125:  //    public:
126:  //        PotentiometerDebugger();
127:  //        virtual ~PotentiometerDebugger();
128:  //
129:  //        /**
130:  //         * @param: lv_obj_t* parent -> parent of the tab
131:  //         * @return: None
132:  //         *
133:  //         * objects are initially loaded onto a NULL parent to be updated later
134:  //         * this sets it so that the parent of the objects is now the tab
135:  //         */
136:  //        void PotentiometerDebuggerInit(lv_obj_t *parent);
137:  // };
138:  //
139:  //
140:  //
141:  // /**
142:  //  * @see: ../Styles.
143:  //  *
144:  //  * show value for limit switch
145:  //  */
146:  // class LimitSwitchDebugger :
147:  //     virtual Styles
148:  // {
149:  // private:
150:  //     lv_obj_t *container;
151:  //
152:  //     lv_obj_t *title1;
153:  //     lv_obj_t *title2;
154:  //
155:  //     lv_obj_t *info1;
156:  //     lv_obj_t *info2;
157:  //
158:  //     protected:
159:  //         /**
160:  //         * @return: None
161:  //         *
162:  //         * updates value of limit switch
163:  //         */
164:  //         void update_limit_switch_info();
165:  //
166:  //     public:
167:  //         LimitSwitchDebugger();
168:  //         virtual ~LimitSwitchDebugger();
169:  //
170:  //         /**
171:  //         * @param: lv_obj_t* parent -> parent of the tab
172:  //         * @return: None
173:  //         *
174:  //         * objects are initially loaded onto a NULL parent to be updated later
175:  //         * this sets it so that the parent of the objects is now the tab
176:  //         */
177:  //         void LimitSwitchDebuggerInit(lv_obj_t *parent);
178:  // };
179:  //
180:  //
181:  //
182:  // /**
183:  //  * @see: ../Styles.
184:  //  *
185:  //  * starts new page with debugger info for vision sensor because it needs more room
186:  //  */
187:  // class VisionSensorDebugger : virtual Styles
188:  // {
189:  //     private:
190:  //         lv_obj_t *title_label;
191:  //
192:  //         lv_obj_t *vision_sensor_screen;
193:  //
194:  //         //back button
195:  //         lv_obj_t *btn_back;
196:  //         lv_obj_t *btn_back_label;
197:  //
198:  //         /**
199:  //         * @param: lv_obj_t* btn -> button that called the funtion
200:  //         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
201:  //         *
202:  //         * button callback function used to go back from the new screen loaded by
203:  //         * this tab because it is predicted to need more space
204:  //         */
205:  //         static lv_res_t btn_back_action(lv_obj_t *btn);
206:  //
207:  //         static bool cont;
208:  //
209:  //     protected:
210:  //         /**
211:  //         * @return: None
212:  //         *
213:  //         * loads a new page with debug info
214:  //         */
215:  //         void load_vision_sensor_page();
216:  //
217:  //     public:
218:  //         VisionSensorDebugger();
219:  //         virtual ~VisionSensorDebugger();
220:  // };
221:  //
222:
223:
224:
225:  //derived class
226:
```

```
227:
228:  /**
229:   * @see: ../Styles.
230:   *
231:   * starts tab object with all the sensor tabs that the user
232:   * can switch between
233:   */
234:  class SensorsDebug :
235:      virtual private Styles
236:  {
237:      private:
238:          //screen
239:          lv_obj_t *sensors_debug_screen;
240:
241:          //title label
242:          lv_obj_t *title_label;
243:
244:          //back button
245:          lv_obj_t *btn_back;
246:          lv_obj_t *btn_back_label;
247:
248:          /**
249:           * @param: lv_obj_t* btn -> button that called the funtion
250:           * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
251:           *
252:           * button callback function used to go back from the debug screen to
253:           * the title screen
254:           */
255:          static lv_res_t btn_back_action(lv_obj_t *btn);
256:
257:
258:          static lv_obj_t *tabview; //tabview object
259:
260:          lv_obj_t *imes_tab;   //individual tabs
261:          lv_obj_t *analog_in_tab;
262:          lv_obj_t *digital_in_tab;
263:          lv_obj_t *imu_tab;
264:          lv_obj_t *encoders_tab;
265:          lv_obj_t *vision_sensor_tab;
266:
267:
268:      public:
269:          SensorsDebug();
270:          ~SensorsDebug();
271:
272:          static bool all_cont; //checks whether to allow user to
273:                                //cycle through tabs or not
274:
275:
276:          /**
277:           * @return: None
278:           *
279:           * contains methods for transition between tabs with checking sensors
280:           * for if they are calibrated or not
281:           * waits for user to go back in a loop while also switching tabs
282:           *
283:           */
284:          void debug();
285:
286:  };
287:
288:
289:
290:
291:  #endif
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/SensorsDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: SensorsDebug.hpp
 *
 * contains all methods for tabs that contain ways to debug and check sensors
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "../Gimmicks.hpp"
#include "../../motors/Motors.hpp"
#include "../../sensors/Sensors.hpp"
#include "SensorsDebug.hpp"
#include "sensor_tabs/AnalogInTab.hpp"
#include "sensor_tabs/DigitalInTab.hpp"
#include "sensor_tabs/EncoderTab.hpp"
#include "sensor_tabs/IMETab.hpp"
#include "sensor_tabs/IMUTab.hpp"
#include "sensor_tabs/VisionSensorTab.hpp"

//base classes
// bool VisionSensorDebugger::cont = true;
bool SensorsDebug::all_cont = true;
lv_obj_t *SensorsDebug::tabview;


SensorsDebug::SensorsDebug()
{
//set default for statics
    all_cont = true;

//init screen
    sensors_debug_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(sensors_debug_screen, &gray);

//init title label
    title_label = lv_label_create(sensors_debug_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, SENSORS_CONTAINER_WIDTH);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Sensors - Debug");

//init tabview
    tabview = lv_tabview_create(sensors_debug_screen, NULL);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BG, &gray);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_REL, &toggle_tabbtn_released);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_PR, &toggle_tabbtn_pressed);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_INDIC, &sw_indic);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_TGL_REL, &toggle_tabbtn_pressed);
    //lv_tabview_set_tab_load_action(tabview, tab_load_action);
    lv_obj_set_width(tabview, SENSORS_CONTAINER_WIDTH);
    lv_obj_set_height(tabview, 200);

//init tabs
    imes_tab = lv_tabview_add_tab(tabview, "IMEs");
    analog_in_tab = lv_tabview_add_tab(tabview, "Analog In");
    digital_in_tab = lv_tabview_add_tab(tabview, "Digital In");
    imu_tab = lv_tabview_add_tab(tabview, "IMU");
    encoders_tab = lv_tabview_add_tab(tabview, "Encoders");
    vision_sensor_tab = lv_tabview_add_tab(tabview, "Vision\nSensor");


//init back button
    //button
    btn_back = lv_btn_create(sensors_debug_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init tabs from other classes
    //IMEsDebuggerInit(imes_tab);
    //PotentiometerDebuggerInit(pot_tab);
    //LimitSwitchDebuggerInit(limit_tab);

//set positions
    lv_obj_set_pos(btn_back, 30, 210);

    lv_obj_set_pos(title_label, 180, 5);

    lv_obj_set_pos(tabview, 20, 25);
}


SensorsDebug::~SensorsDebug()
{
    //deletes widgets instantiated by class
    lv_obj_del(title_label);

    lv_obj_del(btn_back_label);
    lv_obj_del(btn_back);

    lv_obj_del(imes_tab);
    lv_obj_del(analog_in_tab);
    lv_obj_del(digital_in_tab);
    lv_obj_del(vision_sensor_tab);

    lv_obj_del(tabview);

    lv_obj_del(sensors_debug_screen);
```

```
114:  }
115:
116:
117:  /**
118:   * callback funciton that exits main loop when button is pressed
119:   */
120:  lv_res_t SensorsDebug::btn_back_action(lv_obj_t *btn)
121:  {
122:      all_cont = 0;
123:      return LV_RES_OK;
124:  }
125:
126:
127:
128:  /**
129:   * switches on tab loaded, this corresponds to a sensor tab
130:   * if this sensor needs to be calibrated then there is a warning box that
131:   * lets the user choosed to calibrate the sensor, and will not allow the user
132:   * to access the tab until the sensor is calibrated
133:   */
134:  void SensorsDebug::debug()
135:  {
136:      //used to check if user wants to continue cycling through
137:      //tabs. Will be set to zero and loop will break if user hits
138:      //the back button
139:      all_cont = 1;
140:
141:      std::vector<Motor*> v1(Motors::motor_array.begin(), Motors::motor_array.end());
142:      std::vector<std::string> v2(Motors::motor_names_array.begin(), Motors::motor_names_array.end());
143:
144:      IMEsDebugger imes_debug(
145:          imes_tab,
146:          SENSORS_CONTAINER_WIDTH,
147:          SENSORS_CONTAINER_HEIGHT,
148:          v1,
149:          v2
150:      );
151:      AnalogInDebugger analog_in_debug(analog_in_tab, SENSORS_CONTAINER_WIDTH, SENSORS_CONTAINER_HEIGHT, {&Sensors::line_tracker_top, &Sensors::line_tracker_middle, &Sensors::line_tracker_bottom}, {"Tracker Top", "Tracker
Middle", "Tracker Bottom"});
152:      IMUDebugger imu_debug(imu_tab, SENSORS_CONTAINER_WIDTH, SENSORS_CONTAINER_HEIGHT, &Sensors::imu);
153:      EncoderDebugger encoder_debug(
154:          encoders_tab,
155:          SENSORS_CONTAINER_WIDTH,
156:          SENSORS_CONTAINER_HEIGHT,
157:          {
158:              &Sensors::right_encoder,
159:              &Sensors::left_encoder,
160:              &Sensors::strafe_encoder
161:          },
162:          {
163:              "R Encoder",
164:              "L Encoder",
165:              "S Encoder"
166:          }
167:      );
168:
169:      lv_tabview_set_tab_act(tabview, 0, NULL);
170:      lv_scr_load(sensors_debug_screen);
171:
172:      while ( all_cont )
173:      {
174:          switch ( lv_tabview_get_tab_act(tabview) ) //switches to tab user wants to go to
175:          {
176:              case 0:
177:                  imes_debug.update_info();
178:                  break;
179:
180:
181:              case 1:
182:                  analog_in_debug.update_info();
183:                  // if ( !(Sensors::potentiometer.is_calibrated()) )   //checks for sensor being
184:                  //                                                    //calibrated. If not warning
185:                  //                                                    //will appear
186:                  // {
187:                  //     lv_tabview_set_sliding(tabview, false); //dissallows changing
188:                  //                                             //tab until user
189:                  //                                             //has selected a
190:                  //                                             //calibrate option
191:                  //
192:                  //     std::string msg = (
193:                  //         "Potentiometer has not been calibrated.\n"
194:                  //         "Click continue to calibrate, or back to\n"
195:                  //         "return to a previous screen\n\n"
196:                  //         "(Please keep sensor still while calibrating)\n"
197:                  //     );
198:                  //
199:                  //     WarningMessage warnmsg;
200:                  //     bool calibrated = warnmsg.warn(msg, sensors_debug_screen);
201:                  //
202:                  //     lv_tabview_set_sliding(tabview, true); //re-enables switching
203:                  //                                            //tabs
204:                  //
205:                  //     if ( calibrated )
206:                  //     {
207:                  //         Loading load;
208:                  //         load.show_load(500, sensors_debug_screen, 190, 125); //shows loading circle while calibrating
209:                  //         Sensors::potentiometer.calibrate();
210:                  //         load.hide_load();
211:                  //
212:                  //         update_pot_info();
213:                  //     }
214:                  //
215:                  //     else
216:                  //     {
217:                  //         lv_tabview_set_tab_act(tabview, 0, NULL);
218:                  //         //tab_loaded = 0;
219:                  //     }
220:                  // }
221:                  //
222:                  // else //if Accelerometer is already calibrated
223:                  // {
224:                  //     update_pot_info();
225:                  // }
```

```
226:
227:                break;
228:
229:            // case 2:
230:            //     digital_in_debug.update_info();
231:            //     break;
232:            //
233:            // case 3:
234:            //     load_vision_sensor_page();
235:            //     lv_scr_load(sensors_debug_screen);
236:            //
237:            //     //switch to a different tab or user will be unable to leave
238:            //     //vision sensor debugger
239:            //     lv_tabview_set_tab_act(tabview, 0, NULL);
240:            //     //tab_loaded = 0;
241:            //     break;
242:            case 2:
243:                imu_debug.update_info();
244:
245:            case 3:
246:                encoder_debug.update_info();
247:
248:        }
249:
250:        pros::delay(200);
251:    }
252: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/Debug/TitleScreen.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains class that allows user to select a debug tab
8:   *
9:   */
10:
11: #ifndef __TITLESCREEN_HPP__
12: #define __TITLESCREEN_HPP__
13:
14:
15: #include "../../../../include/main.h"
16:
17: #include "../Styles.hpp"
18:
19:
20: /**
21:  * @see: ../Styles.hpp
22:  * @see: Debug.hpp
23:  *
24:  * contains button matrix that has different debug tabs on it
25:  */
26: class TitleScreen : private Styles
27: {
28:     private:
29:         //screen
30:         lv_obj_t *title_screen;
31:
32:         lv_obj_t *title_label;
33:
34:         lv_obj_t *btn_back;
35:         lv_obj_t *btn_back_label;
36:
37:         /**
38:          * @param: lv_obj_t* btn -> button that called the funtion
39:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
40:          *
41:          * button callback function used to set the debug option to -1 meaning the
42:          * user wants to go to the previous screen
43:          */
44:         static lv_res_t btn_back_action(lv_obj_t *btn);
45:
46:
47:         lv_obj_t *button_matrix; //button matrix object
48:         static const char* btnm_map[]; //map for button matrix
49:
50:         /**
51:          * @param: lv_obj_t* btn -> button that called the funtion
52:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
53:          *
54:          * button callback function used to set option of debug screen that user wants to go to
55:          */
56:         static lv_res_t button_matrix_action(lv_obj_t *btnm, const char *btn_txt);
57:
58:
59:     public:
60:         TitleScreen();
61:         ~TitleScreen();
62:
63:         static int option;
64:
65:         /**
66:          * @return: None
67:          *
68:          * @see: btn_back_action
69:          * @see: button_matrix_action
70:          *
71:          * loads screen and waits in a loop with a delay for user to select
72:          * a button
73:          */
74:         void chooseOption();
75:
76: };
77:
78: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/Debug/TitleScreen.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see: TitleScreen.hpp
8:   *
9:   * contains class for selecting a debug screen or going to previous stage
10:  */
11:
12:  #include "../../../../include/main.h"
13:  #include "../../../../include/api.h"
14:
15:  #include "TitleScreen.hpp"
16:  #include "../../controller/controller.hpp"
17:
18:
19:  int TitleScreen::option = 0;
20:  const char* TitleScreen::btnm_map[] = {
21:        "Motors", "Sensors", "Controller", "Battery",
22:        "\n", "Field Control", "Wiring", "Internal\nMotor PID", ""
23:        };
24:
25:  TitleScreen::TitleScreen()
26:  {
27:     option = 0;
28:
29:     title_screen = lv_obj_create(NULL, NULL);
30:     lv_obj_set_style(title_screen, &gray);
31:
32:  //init button matrix
33:     button_matrix = lv_btnm_create(title_screen, NULL);
34:     lv_btnm_set_map(button_matrix, btnm_map);
35:     lv_btnm_set_action(button_matrix, button_matrix_action);
36:     lv_obj_set_width(button_matrix, 440);
37:     lv_obj_set_height(button_matrix, 140);
38:
39:     //set styles of button matrix
40:     lv_btnm_set_style(button_matrix, LV_BTNM_STYLE_BTN_REL, &toggle_btn_released);
41:     lv_btnm_set_style(button_matrix, LV_BTNM_STYLE_BTN_PR, &toggle_btn_pressed);
42:
43:  //init title label
44:     title_label = lv_label_create(title_screen, NULL);
45:     lv_obj_set_style(title_label, &heading_text);
46:     lv_obj_set_width(title_label, 300);
47:     lv_obj_set_height(title_label, 20);
48:     lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
49:     lv_label_set_text(title_label, "Debugger");
50:
51:  //init back button
52:     //button
53:     btn_back = lv_btn_create(title_screen, NULL);
54:     lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
55:     lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
56:     lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
57:     lv_obj_set_width(btn_back, 75);
58:     lv_obj_set_height(btn_back, 25);
59:
60:     //label
61:     btn_back_label = lv_label_create(btn_back, NULL);
62:     lv_obj_set_style(btn_back_label, &heading_text);
63:     lv_label_set_text(btn_back_label, "Back");
64:
65:
66:  //set postitions of widgets
67:     lv_obj_set_pos(btn_back, 210, 200);
68:     lv_obj_set_pos(title_label, 210, 20);
69:     lv_obj_set_pos(button_matrix, 20, 50);
70:
71:  }
72:
73:
74:  TitleScreen::~TitleScreen()
75:  {
76:     lv_obj_del(btn_back_label);
77:     lv_obj_del(btn_back);
78:     lv_obj_del(title_label);
79:     lv_obj_del(button_matrix);
80:
81:     lv_obj_del(title_screen);
82:  }
83:
84:
85:
86:  /**
87:   * compares text of button to text of label to see what button was clicked
88:   * sets int option to value based on the button clicked
89:   */
90:  lv_res_t TitleScreen::button_matrix_action(lv_obj_t *btnm, const char *btn_txt)
91:  {
92:     if (btn_txt == "Motors")
93:     {
94:        option = 1;
95:     }
96:     else if (btn_txt == "Sensors")
97:     {
98:        option = 2;
99:     }
100:    else if (btn_txt == "Controller")
101:    {
102:       option = 3;
103:    }
104:    else if (btn_txt == "Battery")
105:    {
106:       option = 4;
107:    }
108:    else if (btn_txt == "Field Control")
109:    {
110:       option = 5;
111:    }
112:    else if (btn_txt == "Wiring")
113:    {
```

```
114:         option = 6;
115:     }
116:     else if (btn_txt == "Internal\nMotor PID")
117:     {
118:         option = 7;
119:     }
120:     return LV_RES_OK;
121: }
122:
123:
124:
125: /**
126:  * sets option to -1 which is to be interpreted as user wanting to go back
127:  */
128: lv_res_t TitleScreen::btn_back_action(lv_obj_t *btn)
129: {
130:     option = -1;
131:     return LV_RES_OK;
132: }
133:
134:
135:
136: /**
137:  * waits for option to be non zero
138:  * this will happen once any button is clicked
139:  */
140: void TitleScreen::chooseOption()
141: {
142:     Controller controllers;
143:     option = 0;
144:
145:     lv_scr_load(title_screen);
146:     while ( !option )
147:     {
148:         //allow controller to press the buttons as well
149:         if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_B) )
150:         {
151:             btn_back_action( NULL );
152:             pros::delay(100);
153:         }
154:         pros::delay(20);
155:     }
156: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/lcdCode/Debug/Wiring.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 10/15/2019
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains class that shows the current wiring of the robot
8:   */
9:
10:  #ifndef __WIRING_HPP__
11:  #define __WIRING_HPP__
12:
13:  #include "../Styles.hpp"
14:
15:  #include "../../motors/Motors.hpp"
16:  #include "../../sensors/Sensors.hpp"
17:
18:
19:  /**
20:   * @see: ../Styles.hpp
21:   * @see: ../../motors/Motors.hpp
22:   * @see: ../../sensors/Sensors.hpp
23:   *
24:   * shows the ports that each motor or sensor is located on
25:   * purpose is to make it easier and more companct to wire the robot than having
26:   * to read off of separate computer screen
27:   */
28:  class Wiring : private Styles
29:  {
30:      private:
31:          lv_obj_t *wiring_screen;
32:          lv_obj_t *title_label;
33:
34:          lv_obj_t *motor_info;
35:          lv_obj_t *sensors_info;
36:
37:          //back button
38:          lv_obj_t *btn_back;
39:          lv_obj_t *btn_back_label;
40:
41:          /**
42:           * @param: lv_obj_t* btn -> button that called the funtion
43:           * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
44:           *
45:           * button callback function used to set cont to false meaning the
46:           * user wants to go to the title screen
47:           */
48:          static lv_res_t btn_back_action(lv_obj_t *btn);
49:
50:      public:
51:          static bool cont;
52:
53:          Wiring();
54:          ~Wiring();
55:
56:          /**
57:           * @return: None
58:           *
59:           * passive screen -- loads text and wait for user to go back
60:           */
61:          void debug();
62:
63:  };
64:
65:
66:
67:  #endif
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/Wiring.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: Wiring.hpp
 *
 * contains class that shows wiring configuration
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "Wiring.hpp"

#include "../../motors/Motors.hpp"
#include "../../sensors/Sensors.hpp"


bool Wiring::cont = true;

Wiring::Wiring()
{
    Configuration* config = Configuration::get_instance();

    cont = true;

//screen
    wiring_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(wiring_screen, &gray);

//init back button
    //button
    btn_back = lv_btn_create(wiring_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init title label
    title_label = lv_label_create(wiring_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 440);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Wiring");

//init motor info label
    motor_info = lv_label_create(wiring_screen, NULL);
    lv_label_set_style(motor_info, &subheading_text);
    lv_obj_set_width(motor_info, 220);
    lv_obj_set_height(motor_info, 200);
    lv_label_set_align(motor_info, LV_LABEL_ALIGN_LEFT);

    std::string motors_text = (
          "front right    (200 RPM) - " + std::to_string(config->front_right_port) + "\n"
          "back right     (200 RPM) - " + std::to_string(config->back_left_port) + "\n"
          "front left     (200 RPM) - " + std::to_string(config->front_left_port) + "\n"
          "back left      (200 RPM) - " + std::to_string(config->back_right_port) + "\n"
          "left intake    (600 RPM) - " + std::to_string(config->left_intake_port) + "\n"
          "right intake   (600 RPM) - " + std::to_string(config->right_intake_port) + "\n"
          "upper_indexer  (600 RPM) - " + std::to_string(config->upper_indexer_port) + "\n"
          "lower_indexer  (600 RPM) - " + std::to_string(config->lower_indexer_port) + "\n"
    );

    lv_label_set_text(motor_info, motors_text.c_str());

//init motor info label
    sensors_info = lv_label_create(wiring_screen, NULL);
    lv_label_set_style(sensors_info, &subheading_text);
    lv_obj_set_width(sensors_info, 220);
    lv_obj_set_height(sensors_info, 200);
    lv_label_set_align(sensors_info, LV_LABEL_ALIGN_LEFT);

    std::string sensors_text = (
       std::string("right enc top    - ") + RIGHT_ENC_TOP_PORT + "\n" +
       "right enc bottom - " + RIGHT_ENC_BOTTOM_PORT + "\n" +
       "left enc top     - " + LEFT_ENC_TOP_PORT + "\n" +
       "left enc bottom  - " + LEFT_ENC_BOTTOM_PORT + "\n" +
       "potentiometer    - " + POTENTIOMETER_PORT + "\n" +
       "top detector     - " + DETECTOR_TOP_PORT + "\n" +
       "middle detector  - " + DETECTOR_MIDDLE_PORT + "\n" +
       "bottom detector  - " + DETECTOR_BOTTOM_PORT + "\n" +
       "optical sensor   - " + std::to_string(OPTICAL_PORT) + "\n"
    );

    lv_label_set_text(sensors_info, sensors_text.c_str());

//set positions
    lv_obj_set_pos(btn_back, 30, 210);

    lv_obj_set_pos(title_label, 220, 5);

    lv_obj_set_pos(motor_info, 20, 25);
    lv_obj_set_pos(sensors_info, 300, 25);


}

Wiring::~Wiring()
{
    lv_obj_del(wiring_screen);
}
```

```
114:  /**
115:   * sets cont to false signifying user wants to go back, main loop will exit
116:   */
117:  lv_res_t Wiring::btn_back_action(lv_obj_t *btn)
118:  {
119:      cont = false;
120:      return LV_RES_OK;
121:  }
122:
123:
124:  /**
125:   * waits for cont to be false which occurs when the user hits the back button
126:   */
127:  void Wiring::debug()
128:  {
129:      cont = true;
130:
131:      lv_scr_load(wiring_screen);
132:
133:      while ( cont )
134:      {
135:          pros::delay(100);
136:      }
137:  }
```

```
1:  #ifndef __ANALOGINTAB_HPP__
2:  #define __ANALOGINTAB_HPP__
3:
4:  #include <string>
5:  #include <vector>
6:
7:  #include "main.h"
8:
9:  #include "../../../sensors/AnalogInSensor.hpp"
10: #include "../../Styles.hpp"
11:
12:
13: class AnalogInDebugger :
14:     virtual Styles
15: {
16:     private:
17:         lv_obj_t *container;
18:
19:         lv_obj_t *title1;
20:         lv_obj_t *title2;
21:         lv_obj_t *title3;
22:
23:         lv_obj_t *info1;
24:         lv_obj_t *info2;
25:         lv_obj_t *info3;
26:
27:         lv_obj_t *btn_calibrate;
28:         lv_obj_t *btn_calibrate_label;
29:
30:         static std::vector<AnalogInSensor*> sensors;
31:         static std::vector<std::string> names;
32:
33:         /**
34:          * @param: lv_obj_t* btn -> button that called the funtion
35:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
36:          *
37:          * button callback function used to calibrate sensor
38:          */
39:         static lv_res_t btn_calibrate_action(lv_obj_t *btn);
40:
41:     public:
42:         AnalogInDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<AnalogInSensor*> sensors_vec, std::vector<std::string> names_vec);
43:         ~AnalogInDebugger();
44:
45:         /**
46:          * @return: None
47:          *
48:          * updates value of sensors
49:          */
50:         void update_info();
51:
52: };
53:
54:
55:
56: #endif
```

```
#include <string>
#include <vector>

#include "main.h"

#include "../../Gimmicks.hpp"
#include "AnalogInTab.hpp"

std::vector<AnalogInSensor*> AnalogInDebugger::sensors;
std::vector<std::string> AnalogInDebugger::names;


AnalogInDebugger::AnalogInDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<AnalogInSensor*> sensors_vec, std::vector<std::string> names_vec)
{
    for( int i = 0; i < sensors_vec.size(); i++ )
    {
        sensors.push_back(sensors_vec.at(i));
        names.push_back(names_vec.at(i));
    }
//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, x_dim);
    lv_obj_set_height(container, y_dim);

//title for columns
    //1
    title1 = lv_label_create(container, NULL);
    lv_obj_set_style(title1, &toggle_tabbtn_pressed);
    lv_obj_set_width(title1, (x_dim));
    lv_obj_set_height(title1, 20);
    lv_label_set_align(title1, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title1, "None");

    //2
    title2 = lv_label_create(container, NULL);
    lv_obj_set_style(title2, &toggle_tabbtn_pressed);
    lv_obj_set_width(title2, (x_dim/3));
    lv_obj_set_height(title2, 20);
    lv_label_set_align(title2, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title2, "None");

    //3
    title3 = lv_label_create(container, NULL);
    lv_obj_set_style(title3, &toggle_tabbtn_pressed);
    lv_obj_set_width(title3, (x_dim/3));
    lv_obj_set_height(title3, 20);
    lv_label_set_align(title3, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title3, "None");

//info for columns
    //1
    info1 = lv_label_create(container, NULL);
    lv_obj_set_style(info1, &toggle_tabbtn_pressed);
    lv_obj_set_width(info1, (x_dim/3));
    lv_obj_set_height(info1, y_dim - 20);
    lv_label_set_align(info1, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info1, "None");

    //2
    info2 = lv_label_create(container, NULL);
    lv_obj_set_style(info2, &toggle_tabbtn_pressed);
    lv_obj_set_width(info2, (x_dim / 3));
    lv_obj_set_height(info2, y_dim - 20);
    lv_label_set_align(info2, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info2, "None");

    //3
    info3 = lv_label_create(container, NULL);
    lv_obj_set_style(info3, &toggle_tabbtn_pressed);
    lv_obj_set_width(info3, (x_dim/3));
    lv_obj_set_height(info3, y_dim - 20);
    lv_label_set_align(info3, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info3, "None");

//calibrate button
    //button
    btn_calibrate = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_calibrate, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_calibrate, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_calibrate, LV_BTN_ACTION_CLICK, btn_calibrate_action);
    lv_obj_set_width(btn_calibrate, 110);
    lv_obj_set_height(btn_calibrate, 25);

    //label
    btn_calibrate_label = lv_label_create(btn_calibrate, NULL);
    lv_obj_set_style(btn_calibrate_label, &subheading_text);
    lv_label_set_text(btn_calibrate_label, "Calibrate");

//set positions relative to container
    lv_obj_align(title1, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(info1, container, LV_ALIGN_IN_TOP_LEFT, 10, 30);

    lv_obj_align(title2, container, LV_ALIGN_IN_TOP_MID, -15, 10);
    lv_obj_align(info2, container, LV_ALIGN_IN_TOP_MID, -15, 30);

    lv_obj_align(title3, container, LV_ALIGN_IN_TOP_RIGHT, -100, 10);
    lv_obj_align(info3, container, LV_ALIGN_IN_TOP_RIGHT, -100, 30);

    lv_obj_align(btn_calibrate, container, LV_ALIGN_IN_BOTTOM_RIGHT, -50, 0);
}

AnalogInDebugger::~AnalogInDebugger()
{

}


/**
 * calibrates potentiometer and adds loading bar show gui doesn't appear to hang
 */
```

```
114:  lv_res_t AnalogInDebugger::btn_calibrate_action(lv_obj_t *btn)
115:  {
116:      Loading load;
117:      for(int i=0; i < sensors.size(); i++)
118:      {
119:          load.show_load(500, lv_scr_act(), 190, 240); //shows loading bar while calibrating
120:          sensors.at(i)->calibrate();
121:          load.hide_load();
122:      }
123:
124:      return LV_RES_OK;
125:  }
126:
127:
128:
129:  /**
130:   * updates potentiometer data with raw and corrected values
131:   */
132:  void AnalogInDebugger::update_info()
133:  {
134:      std::string names_text = "";
135:      std::string raw_text = "";
136:      std::string corrected_text = "";
137:      for(int i=0; i < sensors.size(); i++)
138:      {
139:          names_text += names.at(i) + "\n";
140:          raw_text += std::to_string(sensors.at(i)->get_raw_value()) + "\n";
141:          corrected_text += std::to_string(sensors.at(i)->get_value(false)) + "\n";
142:      }
143:
144:      lv_label_set_text(title1, "Sensor");
145:      lv_label_set_text(title2, "Raw Input");
146:      lv_label_set_text(title3, "Corrected Input");
147:      lv_label_set_text(info1, names_text.c_str());
148:      lv_label_set_text(info2, raw_text.c_str());
149:      lv_label_set_text(info3, corrected_text.c_str());
150:  }
```

```
1:  #ifndef __DIGITALINTAB_HPP__
2:  #define __DIGITALINTAB_HPP__
3:
4:  #include <string>
5:  #include <vector>
6:
7:  #include "main.h"
8:
9:  #include "../../Styles.hpp"
10:
11:
12: /**
13:  * @see: ../Styles.
14:  *
15:  * show value for limit switch
16:  */
17: class DigitalInDebugger :
18:     virtual Styles
19: {
20: private:
21:     lv_obj_t *container;
22:
23:     lv_obj_t *title1;
24:     lv_obj_t *title2;
25:
26:     lv_obj_t *info1;
27:     lv_obj_t *info2;
28:
29:     static std::vector<pros::ADIDigitalIn*> sensors;
30:     static std::vector<std::string> names;
31:
32:     public:
33:         DigitalInDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<pros::ADIDigitalIn*> sensors_vec, std::vector<std::string> names_vec);
34:         ~DigitalInDebugger();
35:
36:         /**
37:          * @return: None
38:          *
39:          * updates value of limit switch
40:          */
41:         void update_info();
42: };
43:
44:
45: #endif
```

```
1:  #include <string>
2:  #include <vector>
3:
4:  #include "main.h"
5:
6:  #include "../../../motors/Motor.hpp"
7:  #include "../../Styles.hpp"
8:  #include "DigitalInTab.hpp"
9:
10:
11: std::vector<pros::ADIDigitalIn*> DigitalInDebugger::sensors;
12: std::vector<std::string> DigitalInDebugger::names;
13:
14:
15: DigitalInDebugger::DigitalInDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<pros::ADIDigitalIn*> sensors_vec, std::vector<std::string> names_vec)
16: {
17:     for( int i = 0; i < sensors_vec.size(); i++ )
18:     {
19:         sensors.push_back(sensors_vec.at(i));
20:         names.push_back(names_vec.at(i));
21:     }
22:
23: //init container
24:     container = lv_cont_create(parent, NULL);
25:     lv_cont_set_fit(container, false, false);
26:     lv_obj_set_style(container, &gray);
27:     lv_cont_set_fit(container, false, false);
28:     lv_obj_set_width(container, x_dim);
29:     lv_obj_set_height(container, y_dim);
30:
31: //title for columns
32:     //1
33:     title1 = lv_label_create(container, NULL);
34:     lv_obj_set_style(title1, &toggle_tabbtn_pressed);
35:     lv_obj_set_width(title1, (x_dim));
36:     lv_obj_set_height(title1, 20);
37:     lv_label_set_align(title1, LV_LABEL_ALIGN_CENTER);
38:     lv_label_set_text(title1, "None");
39:
40:     //2
41:     title2 = lv_label_create(container, NULL);
42:     lv_obj_set_style(title2, &toggle_tabbtn_pressed);
43:     lv_obj_set_width(title2, (x_dim/3));
44:     lv_obj_set_height(title2, 20);
45:     lv_label_set_align(title2, LV_LABEL_ALIGN_CENTER);
46:     lv_label_set_text(title2, "None");
47:
48: //info for columns
49:     //1
50:     info1 = lv_label_create(container, NULL);
51:     lv_obj_set_style(info1, &toggle_tabbtn_pressed);
52:     lv_obj_set_width(info1, (x_dim/3));
53:     lv_obj_set_height(info1, y_dim - 20);
54:     lv_label_set_align(info1, LV_LABEL_ALIGN_LEFT);
55:     lv_label_set_text(info1, "None");
56:
57:     //2
58:     info2 = lv_label_create(container, NULL);
59:     lv_obj_set_style(info2, &toggle_tabbtn_pressed);
60:     lv_obj_set_width(info2, (x_dim/3));
61:     lv_obj_set_height(info2, y_dim - 20);
62:     lv_label_set_align(info2, LV_LABEL_ALIGN_LEFT);
63:     lv_label_set_text(info2, "None");
64:
65:
66: //set positions relative to container
67:     lv_obj_align(title1, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
68:     lv_obj_align(info1, container, LV_ALIGN_IN_TOP_LEFT, 10, 30);
69:
70:     lv_obj_align(title2, container, LV_ALIGN_IN_TOP_RIGHT, -100, 10);
71:     lv_obj_align(info2, container, LV_ALIGN_IN_TOP_RIGHT, -100, 30);
72:
73: }
74:
75: DigitalInDebugger::~DigitalInDebugger()
76: {
77:
78: }
79:
80:
81: /**
82:  * shows value of limit switch as either 0 or 1
83:  */
84: void DigitalInDebugger::update_info()
85: {
86:     std::string names_text;
87:     std::string val_text;
88:     for(int i=0; i < sensors.size(); i++)
89:     {
90:         names_text += names.at(i) + "\n";
91:         val_text += std::to_string(sensors.at(i)->get_value()) + "\n";
92:     }
93:
94:     lv_label_set_text(title1, "Limit Switch");
95:     lv_label_set_text(title2, "State");
96:     lv_label_set_text(info1, names_text.c_str());
97:     lv_label_set_text(info2, val_text.c_str());
98: }
```

```
1:  #ifndef __ENCODERTAB_HPP__
2:  #define __ENCODERTAB_HPP__
3:
4:  #include <string>
5:  #include <vector>
6:
7:  #include "main.h"
8:
9:  #include "../../../sensors/Encoder.hpp"
10: #include "../../Styles.hpp"
11:
12:
13: class EncoderDebugger :
14:     virtual Styles
15: {
16:     private:
17:         lv_obj_t *container;
18:
19:         lv_obj_t *title1;
20:         lv_obj_t *title2;
21:         lv_obj_t *title3;
22:
23:         lv_obj_t *info1;
24:         lv_obj_t *info2;
25:         lv_obj_t *info3;
26:
27:         lv_obj_t *btn_tare;
28:         lv_obj_t *btn_tare_label;
29:
30:         static std::vector<Encoder*> encoders;
31:         static std::vector<std::string> names;
32:         static std::vector<int> unique_ids;
33:
34:         /**
35:          * @param: lv_obj_t* btn -> button that called the funtion
36:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
37:          *
38:          * button callback function used to tare encoder
39:          */
40:         static lv_res_t btn_tare_action(lv_obj_t *btn);
41:
42:     public:
43:         EncoderDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<Encoder*> encoders_vec, std::vector<std::string> names_vec);
44:         ~EncoderDebugger();
45:
46:         /**
47:          * @return: None
48:          *
49:          * updates value of sensors
50:          */
51:         void update_info();
52:
53: };
54:
55:
56:
57: #endif
```

```
#include <string>
#include <vector>

#include "main.h"

#include "../../Gimmicks.hpp"
#include "EncoderTab.hpp"

std::vector<Encoder*> EncoderDebugger::encoders;
std::vector<std::string> EncoderDebugger::names;
std::vector<int> EncoderDebugger::unique_ids;


EncoderDebugger::EncoderDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<Encoder*> encoders_vec, std::vector<std::string> names_vec)
{
    for( int i = 0; i < encoders_vec.size(); i++ )
    {
        encoders.push_back(encoders_vec.at(i));
        unique_ids.push_back(encoders_vec.at(i)->get_unique_id());
        names.push_back(names_vec.at(i));
    }
  //init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, x_dim);
    lv_obj_set_height(container, y_dim);

  //title for columns
    //1
    title1 = lv_label_create(container, NULL);
    lv_obj_set_style(title1, &toggle_tabbtn_pressed);
    lv_obj_set_width(title1, (x_dim));
    lv_obj_set_height(title1, 20);
    lv_label_set_align(title1, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title1, "None");

    //2
    title2 = lv_label_create(container, NULL);
    lv_obj_set_style(title2, &toggle_tabbtn_pressed);
    lv_obj_set_width(title2, (x_dim/3));
    lv_obj_set_height(title2, 20);
    lv_label_set_align(title2, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title2, "None");

    //3
    title3 = lv_label_create(container, NULL);
    lv_obj_set_style(title3, &toggle_tabbtn_pressed);
    lv_obj_set_width(title3, (x_dim/3));
    lv_obj_set_height(title3, 20);
    lv_label_set_align(title3, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title3, "None");

  //info for columns
    //1
    info1 = lv_label_create(container, NULL);
    lv_obj_set_style(info1, &toggle_tabbtn_pressed);
    lv_obj_set_width(info1, (x_dim/3));
    lv_obj_set_height(info1, y_dim - 20);
    lv_label_set_align(info1, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info1, "None");

    //2
    info2 = lv_label_create(container, NULL);
    lv_obj_set_style(info2, &toggle_tabbtn_pressed);
    lv_obj_set_width(info2, (x_dim / 3));
    lv_obj_set_height(info2, y_dim - 20);
    lv_label_set_align(info2, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info2, "None");

    //3
    info3 = lv_label_create(container, NULL);
    lv_obj_set_style(info3, &toggle_tabbtn_pressed);
    lv_obj_set_width(info3, (x_dim/3));
    lv_obj_set_height(info3, y_dim - 20);
    lv_label_set_align(info3, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info3, "None");

  //calibrate button
    //button
    btn_tare = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_tare, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_tare, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_tare, LV_BTN_ACTION_CLICK, btn_tare_action);
    lv_obj_set_width(btn_tare, 110);
    lv_obj_set_height(btn_tare, 25);

    //label
    btn_tare_label = lv_label_create(btn_tare, NULL);
    lv_obj_set_style(btn_tare_label, &subheading_text);
    lv_label_set_text(btn_tare_label, "Calibrate");

  //set positions relative to container
    lv_obj_align(title1, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(info1, container, LV_ALIGN_IN_TOP_LEFT, 10, 30);

    lv_obj_align(title2, container, LV_ALIGN_IN_TOP_MID, -50, 10);
    lv_obj_align(info2, container, LV_ALIGN_IN_TOP_MID, -50, 30);

    lv_obj_align(title3, container, LV_ALIGN_IN_TOP_RIGHT, -100, 10);
    lv_obj_align(info3, container, LV_ALIGN_IN_TOP_RIGHT, -100, 30);

    lv_obj_align(btn_tare, container, LV_ALIGN_IN_BOTTOM_RIGHT, -50, 0);
}

EncoderDebugger::~EncoderDebugger()
{

}


/**
```

```
114:  * calibrates potentiometer and adds loading bar show gui doesn't appear to hang
115:  */
116: lv_res_t EncoderDebugger::btn_tare_action(lv_obj_t *btn)
117: {
118:    Loading load;
119:    for(int i=0; i < encoders.size(); i++)
120:    {
121:       load.show_load(500, lv_scr_act(), 190, 240); //shows loading bar while calibrating
122:       encoders.at(i)->reset(unique_ids.at(i));
123:       load.hide_load();
124:    }
125:
126:    return LV_RES_OK;
127: }
128:
129:
130:
131: /**
132:  * updates potentiometer data with raw and corrected values
133:  */
134: void EncoderDebugger::update_info()
135: {
136:    std::string names_text = "";
137:    std::string raw_text = "";
138:    std::string corrected_text = "";
139:    for(int i=0; i < encoders.size(); i++)
140:    {
141:       names_text += names.at(i) + "\n";
142:       raw_text += std::to_string(encoders.at(i)->get_absolute_position(false)) + "/" + std::to_string(encoders.at(i)->get_absolute_position(true)) + "\n";
143:       corrected_text += std::to_string(encoders.at(i)->get_position(unique_ids.at(i))) + "\n";
144:    }
145:
146:    lv_label_set_text(title1, "Encoder");
147:    lv_label_set_text(title2, "Absolute Position");
148:    lv_label_set_text(title3, "Unique Position");
149:    lv_label_set_text(info1, names_text.c_str());
150:    lv_label_set_text(info2, raw_text.c_str());
151:    lv_label_set_text(info3, corrected_text.c_str());
152: }
```

```
1:  #ifndef __IMETAB_HPP__
2:  #define __IMETAB_HPP__
3:
4:  #include <string>
5:  #include <vector>
6:
7:  #include "main.h"
8:
9:  #include "../../Styles.hpp"
10: #include "../../../motors/Motor.hpp"
11:
12: /**
13:  * @see: ../Styles.hpp
14:  *
15:  * shows tab of IMEs and allows user to tare encoders and see values
16:  */
17: class IMEsDebugger :
18:     virtual Styles
19: {
20:     private:
21:         lv_obj_t *container;
22:         lv_obj_t *title;
23:         lv_obj_t *info;
24:
25:         lv_obj_t *btn_tare;
26:         lv_obj_t *btn_tare_label;
27:
28:         static std::vector<Motor*> motors;
29:         static std::vector<std::string> names;
30:
31:         /**
32:          * @param: lv_obj_t* btn -> button that called the funtion
33:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
34:          *
35:          * button callback function used to tare all IMEs
36:          */
37:         static lv_res_t btn_tare_action(lv_obj_t *btn);
38:
39:     public:
40:         IMEsDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<Motor*> motors_vec, std::vector<std::string> names_vec);
41:         ~IMEsDebugger();
42:
43:         /**
44:          * @return: None
45:          *
46:          * updates values for IMEs
47:          */
48:         void update_info();
49:
50:
51: };
52:
53:
54: #endif
```

```
1:  #include <string>
2:  #include <vector>
3:
4:  #include "main.h"
5:
6:  #include "../../../motors/Motor.hpp"
7:  #include "../../Styles.hpp"
8:  #include "IMETab.hpp"
9:
10:
11: std::vector<Motor*> IMEsDebugger::motors;
12: std::vector<std::string> IMEsDebugger::names;
13:
14: IMEsDebugger::IMEsDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<Motor*> motors_vec, std::vector<std::string> names_vec)
15: {
16:     for( int i = 0; i < motors_vec.size(); i++ )
17:     {
18:         motors.push_back(motors_vec.at(i));
19:         names.push_back(names_vec.at(i));
20:     }
21:
22:  //init container
23:     container = lv_cont_create(parent, NULL);
24:     lv_cont_set_fit(container, false, false);
25:     lv_obj_set_style(container, &gray);
26:     lv_cont_set_fit(container, false, false);
27:     lv_obj_set_width(container, x_dim);
28:     lv_obj_set_height(container, y_dim);
29:
30:  //default text
31:     std::string text = (
32:         "front right  -\n"
33:         "back right   -\n"
34:         "front left   -\n"
35:         "back left    -\n"
36:         "right lift   -\n"
37:         "left lift    -\n"
38:         "intake       -\n"
39:         "lift         - "
40:     );
41:
42:  //init integrated motor encoders label label
43:     info = lv_label_create(container, NULL);
44:     lv_obj_set_style(info, &toggle_tabbtn_pressed);
45:     lv_obj_set_width(info, (x_dim));
46:     lv_obj_set_height(info, y_dim);
47:     lv_label_set_align(info, LV_LABEL_ALIGN_LEFT);
48:     lv_label_set_text(info, text.c_str());
49:
50:
51:  //init tare encoders button
52:     //button
53:     btn_tare = lv_btn_create(container, NULL);
54:     lv_btn_set_style(btn_tare, LV_BTN_STYLE_REL, &toggle_btn_released);
55:     lv_btn_set_style(btn_tare, LV_BTN_STYLE_PR, &toggle_btn_pressed);
56:     lv_btn_set_action(btn_tare, LV_BTN_ACTION_CLICK, btn_tare_action);
57:     lv_obj_set_width(btn_tare, 110);
58:     lv_obj_set_height(btn_tare, 25);
59:
60:     //label
61:     btn_tare_label = lv_label_create(btn_tare, NULL);
62:     lv_obj_set_style(btn_tare_label, &subheading_text);
63:     lv_label_set_text(btn_tare_label, "tare encoders");
64:
65:
66:  //align objects on container
67:     lv_obj_set_pos(info, 10, 0);
68:     lv_obj_set_pos(btn_tare, 300, (y_dim - 30));
69: }
70:
71: IMEsDebugger::~IMEsDebugger()
72: {
73:
74: }
75:
76:
77: /**
78:  * tares encodes of all motors
79:  */
80: lv_res_t IMEsDebugger::btn_tare_action(lv_obj_t *btn)
81: {
82:     for( int i = 0; i < motors.size(); i++ )
83:     {
84:         motors.at(i)->tare_encoder();
85:     }
86:
87:     return LV_RES_OK;
88: }
89:
90:
91: /**
92:  * updates for each motor to current values
93:  */
94: void IMEsDebugger::update_info()
95: {
96:     int max_characters = 0;
97:     for( int i = 0; i < names.size(); i++ )
98:     {
99:         if(names.at(i).length() > max_characters)
100:        {
101:            max_characters = names.at(i).length();
102:        }
103:    }
104:
105:    std::string text;
106:    for( int i = 0; i < names.size(); i++ )
107:    {
108:        std::string spaces = "";
109:        for(int j = names.at(i).length(); j < max_characters + 3; j++)
110:        {
111:            spaces += " ";
112:        }
113:        text += names.at(i) + spaces + "- " + std::to_string(motors.at(i)->get_encoder_position()) + "\n";
```

```
114:     }
115:
116:     lv_label_set_text(info, text.c_str());
117: }
```

```
1:  #ifndef __IMUTAB_HPP__
2:  #define __IMUTAB_HPP__
3:
4:  #include <string>
5:  #include <vector>
6:
7:  #include "main.h"
8:
9:  #include "../../../sensors/AnalogInSensor.hpp"
10: #include "../../Styles.hpp"
11:
12:
13: class IMUDebugger :
14:     virtual Styles
15: {
16:     private:
17:         lv_obj_t *container;
18:
19:         lv_obj_t *info1;
20:         lv_obj_t *info2;
21:
22:         lv_obj_t *btn_calibrate;
23:         lv_obj_t *btn_calibrate_label;
24:
25:         static pros::Imu *imu;
26:
27:         /**
28:          * @param: lv_obj_t* btn -> button that called the funtion
29:          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
30:          *
31:          * button callback function used to calibrate sensor
32:          */
33:         static lv_res_t btn_calibrate_action(lv_obj_t *btn);
34:
35:     public:
36:         IMUDebugger(lv_obj_t *parent, int x_dim, int y_dim, pros::Imu *imu_sensor);
37:         ~IMUDebugger();
38:
39:         /**
40:          * @return: None
41:          *
42:          * updates value of sensors
43:          */
44:         void update_info();
45:
46: };
47:
48:
49:
50: #endif
```

```cpp
#include <string>
#include <vector>

#include "main.h"

#include "../../Gimmicks.hpp"
#include "IMUTab.hpp"

pros::Imu *IMUDebugger::imu;

IMUDebugger::IMUDebugger(lv_obj_t *parent, int x_dim, int y_dim, pros::Imu *imu_sensor)
{
    imu = imu_sensor;

//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, x_dim);
    lv_obj_set_height(container, y_dim);

//info for columns
    //1
    info1 = lv_label_create(container, NULL);
    lv_obj_set_style(info1, &toggle_tabbtn_pressed);
    lv_obj_set_width(info1, (x_dim/3));
    lv_obj_set_height(info1, y_dim - 20);
    lv_label_set_align(info1, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info1, "None");

    //2
    info2 = lv_label_create(container, NULL);
    lv_obj_set_style(info2, &toggle_tabbtn_pressed);
    lv_obj_set_width(info2, (x_dim / 3));
    lv_obj_set_height(info2, y_dim - 20);
    lv_label_set_align(info2, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info2, "None");

//calibrate button
    //button
    btn_calibrate = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_calibrate, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_calibrate, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_calibrate, LV_BTN_ACTION_CLICK, btn_calibrate_action);
    lv_obj_set_width(btn_calibrate, 110);
    lv_obj_set_height(btn_calibrate, 25);

    //label
    btn_calibrate_label = lv_label_create(btn_calibrate, NULL);
    lv_obj_set_style(btn_calibrate_label, &subheading_text);
    lv_label_set_text(btn_calibrate_label, "Calibrate");

//set positions relative to container
    lv_obj_align(info1, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(info2, container, LV_ALIGN_IN_TOP_MID, -15, 10);

    lv_obj_align(btn_calibrate, container, LV_ALIGN_IN_BOTTOM_RIGHT, -50, 0);
}

IMUDebugger::~IMUDebugger()
{

}


/**
 * calibrates potentiometer and adds loading bar show gui doesn't appear to hang
 */
lv_res_t IMUDebugger::btn_calibrate_action(lv_obj_t *btn)
{
    Loading load;

    load.show_load(2000, lv_scr_act(), 190, 240); //shows loading bar while calibrating
    if(!(imu->get_status() && 0xFF))
    {
        imu->reset();
        while(imu->is_calibrating())
        {
            pros::delay(25);
        }
        load.hide_load();
    }
    return LV_RES_OK;
}


/**
 * updates potentiometer data with raw and corrected values
 */
void IMUDebugger::update_info()
{
    std::string names_text;
    std::string data_text;

    names_text += "heading\nrotation\npitch\nroll\nyaw\naccel x\naccel y\naccel z";
    data_text += std::to_string(imu->get_heading()) + "\n";
    data_text += std::to_string(imu->get_rotation()) + "\n";
    data_text += std::to_string(imu->get_pitch()) + "\n";
    data_text += std::to_string(imu->get_roll()) + "\n";
    data_text += std::to_string(imu->get_yaw()) + "\n";
    data_text += std::to_string(imu->get_accel().x) + "\n";
    data_text += std::to_string(imu->get_accel().y) + "\n";
    data_text += std::to_string(imu->get_accel().z) + "\n";

    lv_label_set_text(info1, names_text.c_str());
    lv_label_set_text(info2, data_text.c_str());
}
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/serial/Logger.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 2/9/2020
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * contains class for a writer queue that accepts writes and flushes them
8:   * to an output stream
9:   */
10:
11: #ifndef __LOGGER_HPP__
12: #define __LOGGER_HPP__
13:
14: #include <atomic>
15: #include <queue>
16: #include <string>
17:
18:
19: typedef struct
20: {
21:     std::string stream;
22:     std::string content;
23: } log_entry;
24:
25:
26: /**
27:  * Contains a queue that can be added to and dumped out so that data can
28:  * be gathered and exported
29:  *
30:  */
31: class Logger
32: {
33:     private:
34:         static std::queue<log_entry> logger_queue;
35:         static std::atomic<bool> lock;
36:         static bool use_queue;
37:
38:         /**
39:          * @param: num_entries -> max number of entries to get
40:          * @return: std::vector<log_entry> -> the list of items gotton from queue
41:          *
42:          * gets an object from the logger queue
43:          * returns an empty string if the queue is empty
44:          */
45:         std::vector<log_entry> get_entries(int num_entries);
46:
47:         /**
48:          * @param: log_entry contents -> what to log
49:          * @return: bool -> true if the file was actually written to, false if an error occured
50:          *
51:          * sends an entry on a given stream
52:          * currently supports cout, clog, and cerr
53:          */
54:         bool log( log_entry entry );
55:
56:     public:
57:         Logger();
58:         ~Logger();
59:
60:         /**
61:          * @param: log_entry test_item -> item to add to the writer queue
62:          * @return: bool -> true on success and false if an error occured in the process
63:          *
64:          * adds an item to the logger queue
65:          * the queue is protected using a spinlock implemented with an
66:          * std::atomic bool
67:          */
68:         bool add( log_entry entry );
69:
70:         /**
71:          * @return: None
72:          *
73:          * builds up a cache of items from the queue for 50ms so that they can be
74:          * logged at closer to the max speed
75:          */
76:         void dump( );
77:
78:         static void stop_queueing();
79:         static void start_queueing();
80:
81:
82:
83:         /**
84:          * @return: int -> number of items in the logger queue
85:          *
86:          * returns the size of the logger queue
87:          */
88:         static int get_count();
89: };
90:
91:
92:
93: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/serial/Logger.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 2/9/2020
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * @see Logger.hpp
8:   *
9:   * contains implementation for the logger class
10:  */
11:
12:  #include <atomic>
13:  #include <iostream>
14:  #include <queue>
15:  #include <string>
16:  #include <vector>
17:
18:  #include "main.h"
19:
20:  #include "Logger.hpp"
21:
22:  std::queue<log_entry> Logger::logger_queue;
23:  std::atomic<bool> Logger::lock = ATOMIC_VAR_INIT(false);
24:  bool Logger::use_queue = true;
25:
26:
27:  Logger::Logger() { }
28:
29:
30:
31:
32:  Logger::~Logger() { }
33:
34:
35:
36:
37:  /**
38:   * fsends data on the given stream based on the log entry
39:   */
40:  bool Logger::log( log_entry entry )
41:  {
42:    if ( entry.stream == "cout" )
43:    {
44:      std::cout << pros::millis() << " " << entry.content << "\n";
45:    }
46:    else if ( entry.stream == "cerr" )
47:    {
48:      std::cerr << pros::millis() << " " << entry.content << "\n";
49:    }
50:    else if ( entry.stream == "clog" )
51:    {
52:      std::clog << pros::millis() << " " << entry.content << "\n";
53:    }
54:    else
55:    {
56:      return false;
57:    }
58:
59:    return true;
60:  }
61:
62:
63:
64:
65:  /**
66:   * add item to the queue by aquiring and releasing atomic lock
67:   */
68:  bool Logger::add( log_entry entry )
69:  {
70:    if ( !entry.stream.empty() && !entry.content.empty() )
71:    {
72:      if(use_queue) {  // save the message in a queue to be viewed later
73:        while ( lock.exchange( true ) ); //aquire lock
74:        logger_queue.push( entry );
75:        lock.exchange( false ); //release lock
76:      } else {  // log the message right away
77:        log(entry);
78:      }
79:
80:      return true;
81:    }
82:
83:    return false;
84:  }
85:
86:
87:
88:
89:  /**
90:   * gets an item from the queue by acquiring the lock and releasing it
91:   */
92:  std::vector<log_entry> Logger::get_entries(int num_entries)
93:  {
94:    std::vector<log_entry> contents;
95:
96:    while ( lock.exchange( true ) ); //aquire lock
97:
98:    for(int i=0; i<num_entries; i++) {
99:      if ( !logger_queue.empty() ) {
100:       contents.push_back(logger_queue.front());
101:       logger_queue.pop();
102:     } else {
103:       break;
104:     }
105:   }
106:
107:   lock.exchange( false ); //release lock because there is no more iteraction
108:                           //with the queue
109:   return contents;
110: }
111:
112:
113:
```

```
114:
115:  /**
116:   * builds up a cache of items
117:   * this is used so that data can be sent at closer to the max speed
118:   */
119:  void Logger::dump( )
120:  {
121:      std::vector<log_entry> entries = get_entries(50);
122:
123:      for ( int i = 0; i < entries.size(); i++ )
124:      {
125:          log(entries.at(i));
126:      }
127:
128:  }
129:
130:
131:  void Logger::start_queueing() {
132:      use_queue = true;
133:  }
134:
135:  void Logger::stop_queueing() {
136:      use_queue = false;
137:  }
138:
139:
140:  /**
141:   * gets the size of the writer queue
142:   */
143:  int Logger::get_count()
144:  {
145:      return logger_queue.size();
146:  }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/serial/Server.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains class for a server that works over serial communication
8:   */
9:
10: #ifndef __SERVER_HPP__
11: #define __SERVER_HPP__
12:
13: #include <atomic>
14: #include <queue>
15: #include <cstdint>
16:
17:
18: typedef struct
19: {
20:     uint16_t return_id;
21:     uint16_t command_id;
22:     std::string msg;
23: } server_request;
24:
25: class Server
26: {
27:     private:
28:         static std::atomic<bool> lock;
29:         static std::queue<server_request> request_queue;
30:
31:         static pros::Task *read_thread;  // the thread for reading stdin
32:
33:         static void read_stdin(void*);
34:
35:         static int num_instances;
36:         static bool debug;
37:
38:         static int delay;
39:
40:         int handle_request(server_request request);
41:
42:     public:
43:         Server();
44:         ~Server();
45:
46:         /**
47:          * @return: None
48:          *
49:          * starts the thread or resmes it if it was stopped
50:          */
51:         void start_server();
52:
53:         /**
54:          * @return: None
55:          *
56:          * stops the thread from being scheduled
57:          */
58:         void stop_server();
59:
60:         void set_server_task_priority(int new_prio);
61:
62:         void set_debug_mode(bool debug_mode);
63:
64:         void clear_stdin();
65:
66:         int handle_requests(int max_requests=10);
67: };
68:
69:
70: #endif
```

```
/**
 * @file: ./RobotCode/src/objects/serial/Server.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains implementation for server implementation
 */
#include <atomic>
#include <cstdint>
#include <queue>
#include <string>

#include "main.h"
#include "pros/apix.h"

#include "../../Configuration.hpp"
#include "../motors/Motors.hpp"
#include "../motors/MotorThread.hpp"
#include "Logger.hpp"
#include "Server.hpp"

std::queue<server_request> Server::request_queue;
std::atomic<bool> Server::lock = ATOMIC_VAR_INIT(false);
pros::Task *Server::read_thread = NULL;
int Server::num_instances = 0;
bool Server::debug = false;
int Server::delay = 100;


Server::Server() {
    if(read_thread == NULL) {
        read_thread = new pros::Task( read_stdin, (void*)NULL, 2, TASK_STACK_DEPTH_DEFAULT, "server_thread");
        read_thread->suspend();
    }

    num_instances += 1;
}



Server::~Server() {
    std::cout << "destructor called on server\n";
    num_instances -= 1;
    if(num_instances == 0) {
        read_thread->remove();
        delete read_thread;
        read_thread = NULL;
    }
}



void Server::read_stdin(void*) {
    int read_check = 0;
    Logger logger;
    log_entry entry;
    int wait_check = 0;

    while(1) {
        char byte = getchar_unlocked();

        if(debug) {
            entry.stream = "clog";
            entry.content = "[INFO] " + std::to_string(pros::millis()) + " Byte read from stdin: " + byte;
            logger.add(entry);
        }

        if(read_check == 0 && byte == '\xAA') {
            read_check = 1;
        } else if(read_check == 1 && byte == '\x55') {
            read_check = 2;
        } else if(read_check == 2 && byte == '\x1E') {
            read_check = 3;
        } else if(read_check == 3) {
            std::string msg;
            int len_msg = (int)byte - 4;  // byte read will be length of bytes to follow
                                          // subtract 4 because next four bytes are handled different
                                          // because they are identifiers
            uint8_t return_msb = getchar_unlocked();
            uint8_t return_lsb = getchar_unlocked();
            uint8_t command_msb = getchar_unlocked();
            uint8_t command_lsb = getchar_unlocked();

            uint16_t return_id = (return_msb << 8) | return_lsb;
            uint16_t command_id = (command_msb << 8) | command_lsb;

            for(int i=0; i<len_msg; i++) {  // read rest message
                msg.push_back(getchar_unlocked());
            }

            char checksum = getchar_unlocked();  // checksum is directly after end of message

            if(debug) {
                entry.stream = "clog";
                entry.content = (
                    "[INFO], "
                    + std::to_string(pros::millis())
                    + ", Return ID read: " + std::to_string(return_id)
                    + ", Command ID read: " + std::to_string(command_id)
                    + ", Msg read: " + msg
                    + ", Checksum read: " + checksum
                );
                logger.add(entry);
            }

            if(checksum == '\xC6')
            {
                server_request request;
                request.return_id = return_id;
                request.command_id = command_id;
                request.msg = msg;
```

```
114:
115:                 while ( lock.exchange( true ) ); //aquire lock
116:                 request_queue.push(request);
117:                 msg = '\0';
118:                 lock.exchange( false ); //release lock
119:             }
120:
121:             read_check = 0;
122:             len_msg = 0;
123:             msg[0] = '\0';
124:
125:         } else {
126:             read_check = 0;
127:         }
128:
129:         wait_check += 1;
130:         if(wait_check > 1024) {
131:             wait_check = 0;
132:             pros::delay(10);
133:         }
134:     }
135: }
136:
137:
138:
139: int Server::handle_request(server_request request) {
140:     // cases are defined in commands.ods
141:     Logger logger;
142:     log_entry entry;
143:     entry.stream = "clog";
144:
145:     std::string return_msg;
146:     return_msg.push_back('\xAA');
147:     return_msg.push_back('\x55');
148:     return_msg.push_back('\x1E');
149:
150:     std::string return_msg_body;
151:     int status;
152:
153:     switch(request.command_id) {
154:     // motor interaction post cases
155:         case 45232: {  //0xB0 0xB0  Set voltage
156:             int motor_number = std::stoi(std::to_string(request.msg.at(0)));
157:             request.msg.erase(0);
158:             int voltage = std::stoi(request.msg);
159:
160:             status = Motors::motor_array.at(motor_number)->set_voltage(voltage);
161:         }
162:         break;
163:
164:         case 45233: {  //0xB0 0xB1  Set Slew Rate
165:             int motor_number = std::stoi(std::to_string(request.msg.at(0)));
166:             request.msg.erase(0);
167:             int slew_rate = std::stoi(request.msg);
168:
169:             status = Motors::motor_array.at(motor_number)->set_slew(slew_rate);
170:         }
171:         break;
172:
173:         case 45234: {  //0xB0 0xB2  Set Port
174:             int motor_number = std::stoi(std::to_string(request.msg.at(0)));
175:             request.msg.erase(0);
176:             int port = std::stoi(request.msg);
177:
178:             status = Motors::motor_array.at(motor_number)->set_port(port);
179:         }
180:         break;
181:
182:         case 45235: {  //0xB0 0xB3  Tare IME
183:             int motor_number = std::stoi(std::to_string(request.msg.at(0)));
184:             request.msg.erase(0);
185:
186:             status = Motors::motor_array.at(motor_number)->tare_encoder();
187:         }
188:         break;
189:
190:         case 45236: {  //0xB0 0xB4  Set Brakemode
191:             int motor_number = std::stoi(std::to_string(request.msg.at(0)));
192:             request.msg.erase(0);
193:             pros::motor_brake_mode_e_t new_brake_mode = static_cast<pros::motor_brake_mode_e_t>(std::stoi(request.msg));
194:
195:             status = Motors::motor_array.at(motor_number)->set_brake_mode(new_brake_mode);
196:         }
197:         break;
198:
199:         case 45237: {  //0xB0 0xB5  Set Gearing
200:             int motor_number = std::stoi(std::to_string(request.msg.at(0)));
201:             request.msg.erase(0);
202:             pros::motor_gearset_e_t new_gearing = static_cast<pros::motor_gearset_e_t>(std::stoi(request.msg));
203:
204:             status = Motors::motor_array.at(motor_number)->set_gearing(new_gearing);
205:         }
206:         break;
207:
208:         case 45238: {  //0xB0 0xB6  Set PID
209:             int motor_number = std::stoi(std::to_string(request.msg.at(0)));
210:             request.msg.erase(0);
211:
212:             char buffer1[8];
213:             char buffer2[8];
214:             char buffer3[8];
215:             char buffer4[8];
216:
217:             std::copy(request.msg.begin(), request.msg.begin() + 8, buffer1);
218:             std::copy(request.msg.begin() + 8, request.msg.begin() + 16, buffer2);
219:             std::copy(request.msg.begin() + 16, request.msg.begin() + 24, buffer3);
220:             std::copy(request.msg.begin() + 24, request.msg.begin() + 32, buffer4);
221:
222:             double n1 = *reinterpret_cast<double*>(buffer1);
223:             double n2 = *reinterpret_cast<double*>(buffer2);
224:             double n3 = *reinterpret_cast<double*>(buffer3);
225:             double n4 = *reinterpret_cast<double*>(buffer4);
226:
```

```
227:                pid pid_constants;
228:                pid_constants.kP = n1;
229:                pid_constants.kI = n2;
230:                pid_constants.kD = n3;
231:                pid_constants.I_max = n4;
232:
233:                status = Motors::motor_array.at(motor_number)->set_pid(pid_constants);
234:            }
235:            break;
236:
237:        case 45239: {  //0xB0 0xB7  Reverse Motor
238:                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
239:                request.msg.erase(0);
240:                int reveresed = std::stoi(request.msg);
241:                status = Motors::motor_array.at(motor_number)->reverse_motor();
242:            }
243:            break;
244:        case 45240: {  //0xB0 0xB8  Set Log Level
245:                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
246:                request.msg.erase(0);
247:                int new_log_level = std::stoi(request.msg);
248:                Motors::motor_array.at(motor_number)->set_log_level(new_log_level);
249:                status = 1;
250:            }
251:            break;
252:        case 45241: {  //0xB0 0xB9  Set Slew enabled/disabled
253:                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
254:                request.msg.erase(0);
255:                int enabled = std::stoi(request.msg);
256:                if(enabled) {
257:                    Motors::motor_array.at(motor_number)->enable_slew();
258:                } else {
259:                    Motors::motor_array.at(motor_number)->disable_slew();
260:                }
261:                status = 1;
262:            }
263:            break;
264:
265:
266:    // motor interaction get cases
267:        case 41120: {  // 0xA0 0xA0  Actual Velocity
268:                int motor_number = request.msg.at(0) - 48;
269:                request.msg.erase(0);
270:
271:                status = 1;
272:                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_actual_velocity());
273:            }
274:            break;
275:
276:        case 41121: {  // 0xA0 0xA1  Actual Voltage
277:                int motor_number = request.msg.at(0) - 48;
278:                request.msg.erase(0);
279:
280:                status = 1;
281:                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_actual_voltage());
282:            }
283:            break;
284:
285:        case 41122: {  // 0xA0 0xA2  Current Draw
286:                int motor_number = request.msg.at(0) - 48;
287:                request.msg.erase(0);
288:
289:                status = 1;
290:                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_current_draw());
291:            }
292:            break;
293:
294:        case 41123: {  // 0xA0 0xA3  Encoder Position
295:                int motor_number = request.msg.at(0) - 48;
296:                request.msg.erase(0);
297:
298:                status = 1;
299:                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_encoder_position());
300:            }
301:            break;
302:
303:        case 41124: {  // 0xA0 0xA4  Brakemode
304:                int motor_number = request.msg.at(0) - 48;
305:                request.msg.erase(0);
306:
307:                status = 1;
308:                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_brake_mode());
309:            }
310:            break;
311:
312:        case 41125: {  // 0xA0 0xA5  Gearset
313:                int motor_number = request.msg.at(0) - 48;
314:                request.msg.erase(0);
315:
316:                status = 1;
317:                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_gearset());
318:            }
319:            break;
320:
321:        case 41126: {  // 0xA0 0xA6  Port
322:                int motor_number = request.msg.at(0) - 48;
323:                request.msg.erase(0);
324:
325:                status = 1;
326:                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_port());
327:            }
328:            break;
329:
330:        case 41127: {  // 0xA0 0xA7  PID Constants
331:                int motor_number = request.msg.at(0) - 48;
332:                request.msg.erase(0);
333:
334:                status = 1;
335:                return_msg_body += std::to_string(Motors::motor_array.at(motor_number)->get_pid().kP);
336:                return_msg_body += " " + std::to_string(Motors::motor_array.at(motor_number)->get_pid().kI);
337:                return_msg_body += " " + std::to_string(Motors::motor_array.at(motor_number)->get_pid().kD);
338:                return_msg_body += " " + std::to_string(Motors::motor_array.at(motor_number)->get_pid().I_max);
339:            }
```

```
340:            break;
341:
342:        case 41128: {  // 0xA0 0xA8  Slew Rate
343:            int motor_number = request.msg.at(0) - 48;
344:            request.msg.erase(0);
345:
346:            status = 1;
347:            return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_slew_rate());
348:        }
349:            break;
350:
351:        case 41129: {  // 0xA0 0xA9  Power
352:            int motor_number = request.msg.at(0) - 48;
353:            request.msg.erase(0);
354:
355:            status = 1;
356:            return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_power());
357:        }
358:            break;
359:
360:        case 41130: {  // 0xA0 0xAA  Temperature
361:            int motor_number = request.msg.at(0) - 48;
362:            request.msg.erase(0);
363:
364:            status = 1;
365:            return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_temperature());
366:        }
367:            break;
368:
369:        case 41131: {  // 0xA0 0xAB  Torque
370:            int motor_number = request.msg.at(0) - 48;
371:            request.msg.erase(0);
372:
373:            status = 1;
374:            return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_torque());
375:        }
376:            break;
377:
378:        case 41132: {  // 0xA0 0xAC  Direction
379:            int motor_number = request.msg.at(0) - 48;
380:            request.msg.erase(0);
381:
382:            status = 1;
383:            return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_direction());
384:        }
385:            break;
386:
387:        case 41133: {  // 0xA0 0xAD  Efficiency
388:            int motor_number = request.msg.at(0) - 48;
389:            request.msg.erase(0);
390:
391:            status = 1;
392:            return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_efficiency());
393:        }
394:            break;
395:
396:        case 41134: {  // 0xA0 0xAE  is stopped
397:            int motor_number = request.msg.at(0) - 48;
398:            request.msg.erase(0);
399:
400:            status = 1;
401:            return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->is_stopped());
402:        }
403:            break;
404:
405:        case 41135: {  // 0xA0 0xAF  is reversed
406:            int motor_number = request.msg.at(0) - 48;
407:            request.msg.erase(0);
408:
409:            status = 1;
410:            return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->is_reversed());
411:        }
412:            break;
413:
414:        case 41376: {  // 0xA1 0xA0  is registered
415:            int motor_number = request.msg.at(0) - 48;
416:            request.msg.erase(0);
417:
418:            status = 1;
419:
420:            MotorThread* motor_thread = MotorThread::get_instance();
421:            return_msg_body = std::to_string(motor_thread->is_registered(*Motors::motor_array.at(motor_number)));
422:        }
423:            break;
424:
425:        // encoder interaction post cases
426:        // encoder iteraction get cases
427:
428:        // analog in sensor interaction post cases
429:        // analog in sensor interaction get cases
430:
431:        // imu interaction post cases
432:        // imu interaction get cases
433:
434:        // position tracker post cases
435:        // position tracker get cases
436:
437:        // sd card interaction post cases
438:
439:        // misc
440:        case 43936:  // 0xAB 0xA0  debug
441:            status = 1;
442:            return_msg_body = " debug msg received: " + request.msg;
443:            break;
444:
445:        case 43937:  // 0xAB 0xA1  init server
446:            status = 1;
447:            pros::c::serctl(SERCTL_DISABLE_COBS, NULL);
448:            set_server_task_priority(TASK_PRIORITY_DEFAULT);  // more messages are sure to follow so give read task more CPU time
449:            delay = 10; // lower delay because of expected messages
450:            return_msg_body = "server is running";
451:            break;
452:
```

```
453:         case 43938:  // 0xAB 0xA2  shutdown server
454:             status = 1;
455:             pros::c::serctl(SERCTL_ENABLE_COBS, NULL);
456:             set_server_task_priority(2);
457:             delay = 100;
458:             return_msg_body = "server is no longer running";
459:             break;
460:
461:         default:
462:             status = 1;
463:             return_msg_body = " [INFO], " + std::to_string(pros::millis()) + ", Invalid Command: " + request.msg;
464:             break;
465:
466:
467:     }
468:
469:     return_msg.push_back(return_msg_body.length() + 2);
470:     return_msg.push_back((char)(request.return_id >> 8) & 0xFF);
471:     return_msg.push_back((char)request.return_id & 0xFF);
472:     return_msg += return_msg_body;
473:     // return_msg += std::to_string(pros::millis());
474:     return_msg.push_back('\xC6');
475:
476:     entry.content = return_msg;
477:
478:     logger.add(entry);
479:
480:     return 1;
481: }
482:
483:
484:
485:
486: void Server::start_server() {
487:     read_thread->resume();
488: }
489:
490: void Server::stop_server() {
491:     read_thread->suspend();
492: }
493:
494: void Server::set_server_task_priority(int new_prio) {
495:     read_thread->set_priority(new_prio);
496: }
497:
498: void Server::set_debug_mode(bool debug_mode) {
499:     debug = debug_mode;
500: }
501:
502: void Server::clear_stdin() {
503:     fflush(stdin);
504:     std::cin.clear();
505: }
506:
507:
508:
509: int Server::handle_requests(int max_requests) {
510:     std::vector<server_request> requests;
511:
512:     if ( !request_queue.empty() ) {
513:         while ( lock.exchange( true ) ); //aquire lock
514:         for(int i=0; i<max_requests; i++) {
515:             if ( !request_queue.empty() ) {
516:                 server_request request = request_queue.front();
517:                 request_queue.pop();
518:
519:                 requests.push_back(request);
520:             }
521:         }
522:         lock.exchange( false ); //release lock
523:     }
524:
525:     for (int i=0; i<requests.size(); i++) {
526:         handle_request(requests.at(i));
527:     }
528:
529:     return requests.size();
530: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/position_tracking/PositionTracker.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * contains functions for calculating robot position
8:   *
9:   */
10:
11: #ifndef __POSITIONTRACKER_HPP__
12: #define __POSITIONTRACKER_HPP__
13:
14: #include <atomic>
15:
16: #include "main.h"
17:
18:
19: #define WHEEL_TRACK_R 2.47
20: #define WHEEL_TRACK_L 2.47
21: #define S_ENC_OFFSET 3.5
22:
23: typedef struct
24: {
25:     long double x_pos = 0;
26:     long double y_pos = 0;
27:     long double theta = 0;
28:     void print() {
29:         std::cout << "x pos: " << this->x_pos << "\n";
30:         std::cout << "y pos: " << this->y_pos << "\n";
31:         std::cout << "angle: " << this->theta << "\n";
32:     };
33: } position;
34:
35:
36: class PositionTracker
37: {
38:     private:
39:         PositionTracker();
40:         static PositionTracker *tracker_obj;
41:
42:         static position current_position;
43:
44:         static long double initial_l_enc;
45:         static long double initial_r_enc;
46:         static long double initial_theta;
47:         static long double imu_offset;
48:
49:         static long double prev_l_enc;
50:         static long double prev_r_enc;
51:         static long double delta_theta_rad;
52:
53:         static int l_id;
54:         static int r_id;
55:
56:         static std::atomic<bool> lock;  //protect position from concurrent access
57:
58:         static int log_level;
59:         static bool use_imu;
60:
61:
62:         static void calc_position(void*);
63:         pros::Task *thread;  // the thread for keeping track of position
64:
65:
66:     public:
67:         ~PositionTracker();
68:
69:         /**
70:          * @return: PositionTracker -> instance of class to be used throughout program
71:          *
72:          * give the instance of the singleton class or creates it if it does
73:          * not yet exist
74:          */
75:         static PositionTracker* get_instance();
76:
77:         static long double to_inches( long double encoder_ticks, long double wheel_size );
78:         static long double to_encoder_ticks(long double inches, long double wheel_size);
79:         static long double to_radians(long double degrees);
80:         static long double to_degrees(long double radians);
81:
82:         /**
83:          * @return: None
84:          *
85:          * starts the thread or resmes it if it was stopped
86:          */
87:         void start_thread();
88:
89:         /**
90:          * @return: None
91:          *
92:          * stops the thread from being scheduled
93:          */
94:         void stop_thread();
95:
96:         void set_log_level(int log_lvl);
97:
98:         void enable_imu();
99:         void disable_imu();
100:
101:        long double get_delta_theta_rad();
102:        long double get_heading_rad();
103:
104:        position get_position();
105:
106:        static void set_position(position robot_coordinates);
107: };
108:
109: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/position_tracking/PositionTracker.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   * TODO:
7:   *
8:   * contains implementation for functions that track position
9:   */
10:
11:  #include <atomic>
12:
13:  #include "main.h"
14:
15:  #include "../serial/Logger.hpp"
16:  #include "../sensors/Sensors.hpp"
17:  #include "PositionTracker.hpp"
18:
19:
20:  PositionTracker *PositionTracker::tracker_obj = NULL;
21:  std::atomic<bool> PositionTracker::lock = ATOMIC_VAR_INIT(false);
22:  position PositionTracker::current_position;
23:
24:  long double PositionTracker::initial_l_enc;
25:  long double PositionTracker::initial_r_enc;
26:  long double PositionTracker::initial_theta;
27:  long double PositionTracker::imu_offset;
28:
29:  long double PositionTracker::prev_l_enc;
30:  long double PositionTracker::prev_r_enc;
31:  long double PositionTracker::delta_theta_rad;
32:
33:  int PositionTracker::l_id = -1;  // -1 is used as an invalid id
34:  int PositionTracker::r_id = -1;
35:
36:  int PositionTracker::log_level = 0;
37:  bool PositionTracker::use_imu = false;
38:
39:
40:  PositionTracker::PositionTracker() {
41:      thread = new pros::Task( calc_position, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "position_tracking");
42:      set_position({0, 0, 0});
43:      thread->suspend();
44:  }
45:
46:
47:  PositionTracker::~PositionTracker() {
48:      thread->remove();
49:      delete thread;
50:  }
51:
52:
53:  /**
54:   * inits object if object is not already initialized based on a static bool
55:   * sets bool if it is not set
56:   */
57:  PositionTracker* PositionTracker::get_instance() {
58:      if ( tracker_obj == NULL )
59:      {
60:          tracker_obj = new PositionTracker;
61:      }
62:      return tracker_obj;
63:  }
64:
65:
66:
67:
68:  long double PositionTracker::to_inches( long double encoder_ticks, long double wheel_size ) {
69:      long double circumference = (wheel_size * M_PI);
70:      long double revolutions = encoder_ticks / 360.0;
71:      long double inches = circumference * revolutions;
72:
73:      return inches;
74:  }
75:
76:
77:  long double PositionTracker::to_encoder_ticks(long double inches, long double wheel_size) {
78:      long double circumference = (wheel_size * M_PI);
79:      long double revolutions = inches / circumference;
80:      long double encoder_ticks = revolutions * 360;
81:
82:      return encoder_ticks;
83:  }
84:
85:
86:  long double PositionTracker::to_degrees(long double radians) {
87:      return (radians * (180 / M_PI));
88:  }
89:
90:
91:  long double PositionTracker::to_radians(long double degrees) {
92:      return (degrees * (M_PI / 180));
93:  }
94:
95:
96:
97:  void PositionTracker::calc_position(void*)
98:  {
99:      int s_id = Sensors::strafe_encoder.get_unique_id();
100:
101:     prev_l_enc = std::get<0>(Sensors::get_average_encoders(l_id, r_id));
102:     prev_r_enc = std::get<1>(Sensors::get_average_encoders(l_id, r_id));
103:     long double prev_s_enc = Sensors::strafe_encoder.get_position(s_id);
104:
105:     while(1)
106:     {
107:         while ( lock.exchange( true ) );
108:
109:         long double l_enc = std::get<0>(Sensors::get_average_encoders(l_id, r_id));
110:         long double r_enc = std::get<1>(Sensors::get_average_encoders(l_id, r_id));
111:         long double s_enc = Sensors::strafe_encoder.get_position(s_id);
112:         // std::cout << l_enc << " " << r_enc << " " << s_enc << "\n";
113:         long double delta_l_in = to_inches(l_enc - prev_l_enc, 3.25);  // calculate change in each encoder in inches
```

```
114:     long double delta_r_in = to_inches(r_enc - prev_r_enc, 3.25);
115:     long double delta_s_in = to_inches(s_enc - prev_s_enc, 3.25);
116:
117:     prev_l_enc = l_enc;  // update previous encoder values
118:     prev_r_enc = r_enc;
119:     prev_s_enc = s_enc;
120:
121:     // calculate total change in encoders
122:     long double delta_l_total = to_inches(l_enc, 3.25) - to_inches(initial_l_enc, 3.25);
123:     long double delta_r_total = to_inches(r_enc, 3.25) - to_inches(initial_r_enc, 3.25);
124:     // std::cout << "encoder data: " << delta_l_total << " " << delta_r_total << " " << initial_l_enc << " " << initial_r_enc << "\n";
125:
126:     // calculate absolute orientation (unbounded)
127:     long double encoder_reading_rad = initial_theta + ((delta_l_total - delta_r_total) / (WHEEL_TRACK_L + WHEEL_TRACK_R));  // wheel track length
128:     // wrap angle to [-pi, pi]
129:     encoder_reading_rad = std::atan2(std::sin(encoder_reading_rad), std::cos(encoder_reading_rad));
130:
131:
132:     long double new_abs_theta_rad;
133:     long double imu_reading_rad;
134:     if(use_imu) {
135:       imu_reading_rad = imu_offset + to_radians(Sensors::imu.get_heading());
136:       imu_reading_rad = std::atan2(std::sin(imu_reading_rad), std::cos(imu_reading_rad));  // wrap angle to [-pi, pi]
137:
138:       // make sure that imu_reading and theta from encoders have the same sign
139:       // to ensure that they are telling the same reading when merging
140:       // ie. imu = -359, enc = 1    == bad merge
141:       //     imu = -10,  enc = 2    == good merge
142:       if(encoder_reading_rad > 0 && imu_reading_rad < 0 && std::abs(encoder_reading_rad) + std::abs(imu_reading_rad) > (M_PI / 2)) {
143:         imu_reading_rad += 2 * M_PI;
144:       } else if(encoder_reading_rad < 0 && imu_reading_rad > 0 && std::abs(encoder_reading_rad) + std::abs(imu_reading_rad) > (M_PI / 2)) {
145:         imu_reading_rad -= 2 * M_PI;
146:       }
147:
148:       new_abs_theta_rad = (.7 * imu_reading_rad) + (.3 * encoder_reading_rad);  // merge with imu
149:     } else {
150:       new_abs_theta_rad = encoder_reading_rad;
151:     }
152:
153:     // calculate the change in angle from the previous position
154:     delta_theta_rad = new_abs_theta_rad - current_position.theta;
155:
156:     // calculate local offset
157:     long double delta_local_x;
158:     long double delta_local_y;
159:     if(std::abs(delta_theta_rad) < 0.000001) {
160:       delta_local_x = delta_s_in;
161:       delta_local_y = delta_r_in;  // note: delta_l == delta_r
162:     } else {
163:       delta_local_x = (2 * std::sin((delta_theta_rad / 2))) * ((delta_s_in / delta_theta_rad) + S_ENC_OFFSET);
164:       delta_local_y = (2 * std::sin((delta_theta_rad / 2))) * ((delta_r_in / delta_theta_rad) + WHEEL_TRACK_R);
165:     }
166:
167:     // calculate average orientation for the cycle
168:     double avg_theta_rad = current_position.theta + (delta_theta_rad / 2);
169:
170:     // calculate global change in coordinates as the change in the local offset
171:     // rotated by -(avg_theta_rad)
172:     // Converts to polar coordinates, changes the angle, and converts back to cartesian
173:     long double radius_pol = std::sqrt((std::pow(delta_local_x, 2) + std::pow(delta_local_y, 2)));
174:     long double theta_pol = std::atan2(delta_local_y, delta_local_x);
175:     theta_pol = theta_pol - avg_theta_rad;
176:     long double delta_global_x = radius_pol * std::cos(theta_pol);
177:     long double delta_global_y = radius_pol * std::sin(theta_pol);
178:
179:     if (std::isnan(delta_global_x)) {
180:      delta_global_x = 0;
181:     }
182:
183:     if (std::isnan(delta_global_y)) {
184:      delta_global_y = 0;
185:     }
186:
187:     if (std::isnan(new_abs_theta_rad)) {
188:      new_abs_theta_rad = 0;
189:     }
190:
191:     // don't use built in method to update position because that resets encoders, which is not necessary
192:     current_position.x_pos = current_position.x_pos + delta_global_x;
193:     current_position.y_pos = current_position.y_pos + delta_global_y;
194:     current_position.theta = new_abs_theta_rad;
195:
196:
197:     Logger logger;
198:     log_entry entry;
199:
200:     for(int i = 0; i <= log_level; i++) {
201:       switch(i) {
202:         case 0:
203:           entry.content = "";
204:           break;
205:         case 1:
206:           entry.content += ("[INFO], " + std::string("Position Tracking Data")
207:             + ", Time: " + std::to_string(pros::millis())
208:             + ", X_POS: " + std::to_string(current_position.x_pos)
209:             + ", Y_POS: " + std::to_string(current_position.y_pos)
210:             + ", Angle: " + std::to_string(to_degrees(current_position.theta))
211:           );
212:           break;
213:         case 2:
214:           entry.content += (
215:             "angle_from_imu_radians: " + std::to_string(imu_reading_rad)
216:             + "angle_from_encoders_radians: " + std::to_string(encoder_reading_rad)
217:             + "angle_from_imu_degrees: " + std::to_string(to_degrees(imu_reading_rad))
218:             + "angle_from_encoders_degrees: " + std::to_string(to_degrees(encoder_reading_rad))
219:           );
220:           break;
221:         case 3:
222:           entry.content += (
223:             "local_delta_y: " + std::to_string(delta_local_y)
224:             + "local_delta_x: " + std::to_string(delta_local_x)
225:             + "global_delta_y: " + std::to_string(delta_global_y)
226:             + "global_delta_x: " + std::to_string(delta_global_x)
```

```
227:                 );
228:                 break;
229:             case 4:
230:                 entry.content += (
231:                     "l_enc: " + std::to_string(l_enc)
232:                     + "r_enc: " + std::to_string(r_enc)
233:                     + "s_enc: " + std::to_string(s_enc)
234:                     + "delta_l_enc_in: " + std::to_string(delta_l_in)
235:                     + "delta_r_enc_in: " + std::to_string(delta_r_in)
236:                     + "delta_s_enc_in: " + std::to_string(delta_s_in)
237:                 );
238:                 break;
239:             case 5:
240:                 if(use_imu) {
241:                     entry.content += (
242:                         "imu_reading: " + std::to_string(Sensors::imu.get_heading())
243:                         + "imu_offset: " + std::to_string(imu_offset)
244:                     );
245:                 }
246:                 break;
247:         }
248:     }
249:
250:     entry.stream = "clog";
251:     if(!entry.content.empty()) {
252:         logger.add(entry);
253:     }
254:
255:     lock.exchange(false);
256:
257:     pros::delay(5);
258: }
259: }
260:
261:
262:
263:
264: void PositionTracker::start_thread()
265: {
266:     thread->resume();
267: }
268:
269: void PositionTracker::stop_thread()
270: {
271:     thread->suspend();
272: }
273:
274:
275:
276:
277: void PositionTracker::set_log_level(int log_lvl) {
278:     while ( lock.exchange( true ) );
279:     log_level = log_lvl;
280:     lock.exchange(false);
281: }
282:
283: void PositionTracker::enable_imu() {
284:     while ( lock.exchange( true ) );
285:     use_imu = true;
286:     lock.exchange(false);
287: }
288:
289: void PositionTracker::disable_imu() {
290:     while ( lock.exchange( true ) );
291:     use_imu = false;
292:     lock.exchange(false);
293: }
294:
295:
296: long double PositionTracker::get_delta_theta_rad() {
297:     while ( lock.exchange( true ) );
298:     long double d_theta_rad = delta_theta_rad;
299:     lock.exchange(false);
300:
301:     return d_theta_rad;
302: }
303:
304: long double PositionTracker::get_heading_rad() {
305:     while ( lock.exchange( true ) );
306:     long double heading = current_position.theta;
307:     lock.exchange(false);
308:     return heading;
309: }
310:
311: position PositionTracker::get_position()
312: {
313:     while ( lock.exchange( true ) );
314:     position pos;
315:     pos.x_pos = current_position.x_pos;
316:     pos.y_pos = current_position.y_pos;
317:     pos.theta = current_position.theta;
318:     lock.exchange(false);
319:
320:     return pos;
321: }
322:
323:
324:
325:
326: void PositionTracker::set_position(position robot_coordinates)
327: {
328:     while ( lock.exchange( true ) );
329:
330:     if(l_id != -1) {
331:         Sensors::left_encoder.forget_position(l_id);
332:     }
333:     if(r_id != -1) {
334:         Sensors::right_encoder.forget_position(r_id);
335:     }
336:     l_id = Sensors::left_encoder.get_unique_id(true);
337:     r_id = Sensors::right_encoder.get_unique_id(true);
338:
339:     initial_l_enc = std::get<0>(Sensors::get_average_encoders(l_id, r_id));
```

```
340:     initial_r_enc = std::get<1>(Sensors::get_average_encoders(l_id, r_id));
341:     initial_theta = robot_coordinates.theta;
342:
343:     if(use_imu) {
344:         imu_offset = initial_theta - to_radians(Sensors::imu.get_heading());  // offset + imu_reading = initial_theta
345:     } else {
346:         imu_offset = initial_theta;
347:     }
348:
349:     prev_l_enc = initial_l_enc;
350:     prev_r_enc = initial_r_enc;
351:
352:     delta_theta_rad = 0;
353:
354:     current_position = robot_coordinates;
355:
356:     lock.exchange(false);
357: }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/subsystems/Indexer.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * Contains class for the differential subsystem
8:   * has methods for brake and indexing
9:   */
10:
11: #ifndef __INDEXER_HPP__
12: #define __INDEXER_HPP__
13:
14: #include <tuple>
15: #include <queue>
16:
17: #include "main.h"
18:
19: #include "../motors/Motor.hpp"
20: #include "../sensors/Sensors.hpp"
21: #include "../sensors/BallDetector.hpp"
22:
23:
24:
25: struct ball_positions{
26:     bool top;
27:     bool middle;
28:     std::string middle_color;
29:     bool operator== (const ball_positions &state) {
30:         bool top_equal = (top == state.top);
31:         bool middle_equal = (middle == state.middle);
32:         bool middle_color_equal;
33:         if(middle_color == "present" && state.middle_color != "none") {
34:             middle_color_equal = true;
35:         } else if (middle_color == "any") {
36:             middle_color_equal = true;
37:         }else if (middle_color == "red" && state.middle_color == "red") {
38:             middle_color_equal = true;
39:         } else if (middle_color == "blue" && state.middle_color == "blue") {
40:             middle_color_equal = true;
41:         } else if (middle_color == "none" && state.middle_color == "none") {
42:             middle_color_equal = true;
43:         }
44:
45:         return (top_equal && middle_equal && middle_color_equal);
46:     }
47:     bool operator!= (const ball_positions &state) {
48:         bool top_equal = (top == state.top);
49:         bool middle_equal = (middle == state.middle);
50:         bool middle_color_equal;
51:         if(middle_color == "any" && state.middle_color != "none") {
52:             middle_color_equal = true;
53:         } else if (middle_color == "red" && state.middle_color == "red") {
54:             middle_color_equal = true;
55:         } else if (middle_color == "blue" && state.middle_color == "blue") {
56:             middle_color_equal = true;
57:         } else if (middle_color == "none" && state.middle_color == "none") {
58:             middle_color_equal = true;
59:         }
60:
61:         return !(top_equal && middle_equal && middle_color_equal);
62:     }
63: };
64:
65: typedef enum e_indexer_command {
66:     e_index,
67:     e_filter,
68:     e_auto_index,
69:     e_index_no_backboard,
70:     e_index_until_filtered,
71:     e_increment,
72:     e_auto_increment,
73:     e_index_to_state,
74:     e_fix_ball,
75:     e_run_upper,
76:     e_run_lower,
77:     e_stop
78: } indexer_command;
79:
80: typedef struct {
81:     ball_positions end_state;
82:     bool allow_filter;
83: }indexer_args;
84:
85: typedef struct {
86:     int uid;
87:     indexer_command command;
88:     indexer_args args;
89: } indexer_action;
90:
91: /**
92:  * @see: Motors.hpp
93:  *
94:  * contains methods to allow for control of the indexer
95:  */
96: class Indexer
97: {
98:     private:
99:         static Motor *upper_indexer;
100:        static Motor *lower_indexer;
101:        static BallDetector *ball_detector;
102:        static std::string filter_color;
103:
104:        static int num_instances;
105:
106:        pros::Task *thread; // the motor thread
107:        static std::queue<indexer_action> command_queue;
108:        static std::vector<int> commands_finished;
109:        static std::atomic<bool> command_start_lock;
110:        static std::atomic<bool> command_finish_lock;
111:
112:        int send_command(indexer_command command, indexer_args args={});
113:
```

```
114:        static bool auto_filter_ball();
115:        static void indexer_motion_task(void*);
116:
117:    public:
118:        Indexer(Motor &upper, Motor &lower, BallDetector &detector, std::string color);
119:        ~Indexer();
120:
121:        void index();
122:        void filter();
123:        void auto_index();
124:        void index_no_backboard();
125:        int index_until_filtered(bool asynch=false);
126:        int index_to_state(bool allow_filter, ball_positions end_state, bool asynch=false);
127:
128:        void increment();
129:        void auto_increment();
130:
131:        void run_upper_roller();
132:        void run_lower_roller();
133:
134:        int fix_ball(bool asynch=true);
135:
136:        void hard_stop();
137:        void stop();
138:
139:        static ball_positions get_state();
140:
141:        void reset_command_queue();
142:        void update_filter_color(std::string new_color);
143:
144:        void wait_until_finished(int uid);
145:        bool is_finished(int uid);
146:
147:
148:    };
149:
150:
151:
152:
153:    #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/subsystems/Indexer.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * Contains implementation for the differential subsystem
8:   * has methods for brake and indexing
9:   */
10:
11:  #include "main.h"
12:
13:
14:  #include "../serial/Logger.hpp"
15:  #include "../sensors/BallDetector.hpp"
16:  #include "Indexer.hpp"
17:
18:  int Indexer::num_instances = 0;
19:  std::queue<indexer_action> Indexer::command_queue;
20:  std::vector<int> Indexer::commands_finished;
21:  std::atomic<bool> Indexer::command_start_lock = ATOMIC_VAR_INIT(false);
22:  std::atomic<bool> Indexer::command_finish_lock = ATOMIC_VAR_INIT(false);
23:
24:  Motor* Indexer::upper_indexer;
25:  Motor* Indexer::lower_indexer;
26:  BallDetector* Indexer::ball_detector;
27:  std::string Indexer::filter_color;
28:
29:
30:  Indexer::Indexer(Motor &upper, Motor &lower, BallDetector &detector, std::string color)
31:  {
32:      upper_indexer = &upper;
33:      lower_indexer = &lower;
34:      ball_detector = &detector;
35:      filter_color = color;
36:
37:      upper_indexer->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
38:      lower_indexer->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
39:
40:      upper_indexer->set_motor_mode(e_voltage);
41:      lower_indexer->set_motor_mode(e_voltage);
42:
43:      upper_indexer->disable_slew();
44:      lower_indexer->disable_slew();
45:
46:      if(num_instances == 0 || thread == NULL) {
47:          thread = new pros::Task( indexer_motion_task, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "indexer_thread");
48:      }
49:
50:      num_instances += 1;
51:  }
52:
53:
54:  Indexer::~Indexer() {
55:      num_instances -= 1;
56:      if(num_instances == 0) {
57:          delete thread;
58:      }
59:  }
60:
61:
62:
63:
64:  bool Indexer::auto_filter_ball() {
65:      int color = ball_detector->check_filter_level();
66:      if((color == 1 && filter_color == "blue") || (color == 2 && filter_color == "red")) {  // ball should be filtered
67:          upper_indexer->set_voltage(-12000);
68:          lower_indexer->set_voltage(12000);
69:          pros::delay(250);  // let ball filter out
70:          upper_indexer->set_voltage(0);
71:          lower_indexer->set_voltage(0);
72:
73:          return true;  // did filter out ball
74:      } else if(color < 0) {  // ball was detected but color could not be determined: print error message and default to intaking
75:          Logger logger;
76:          log_entry entry;
77:          entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", ball was detected but color could not be determined";
78:          entry.stream = "cerr";
79:          logger.add(entry);
80:      }
81:
82:      return false;  // did not filter out ball
83:  }
84:
85:
86:
87:
88:  void Indexer::indexer_motion_task(void*) {
89:      while(1) {
90:          if(command_queue.empty()) {  // delay unitl there is a command in the queue
91:              pros::delay(5);
92:              continue;
93:          }
94:
95:          // take lock and get command
96:          while ( command_start_lock.exchange( true ) ); //aquire lock
97:          indexer_action action = command_queue.front();
98:          command_queue.pop();
99:          command_start_lock.exchange( false ); //release lock
100:
101:         // execute command
102:         switch(action.command) {
103:             case e_filter: {
104:                 upper_indexer->set_voltage(-12000);
105:                 lower_indexer->set_voltage(12000);
106:                 break;
107:             } case e_auto_index: {
108:                 auto_filter_ball();
109:                 // fallthrough and index like normal now that it doesn't need to filter
110:             } case e_index: {
111:                 upper_indexer->set_voltage(12000);
112:                 lower_indexer->set_voltage(12000);
113:                 break;
```

```
114:         } case e_index_no_backboard: {
115:           upper_indexer->set_voltage(9000);
116:           lower_indexer->set_voltage(12000);
117:           break;
118:         } case e_index_until_filtered: {
119:           upper_indexer->set_voltage(12000);
120:           lower_indexer->set_voltage(12000);
121:
122:           bool filtered = false;
123:           do {
124:             filtered = auto_filter_ball();
125:           } while(!filtered);
126:
127:           break;
128:         } case e_index_to_state: {
129:           ball_positions current_state = get_state();
130:           do {
131:             current_state = get_state();
132:             int color = ball_detector->check_filter_level();
133:             if(action.args.allow_filter) {
134:               auto_filter_ball(); // attempt to filter
135:             }
136:
137:             if(current_state.top != action.args.end_state.top) {
138:               upper_indexer->set_voltage(12000);
139:             }
140:
141:             if(current_state.middle != action.args.end_state.middle) {
142:               lower_indexer->set_voltage(12000);
143:             }
144:           } while(current_state != action.args.end_state);
145:
146:           break;
147:         } case e_auto_increment: {
148:           // try to filter out ball at second level if necessary
149:           auto_filter_ball();
150:           // fall through if there is nothing to filter out
151:         } case e_increment: {
152:           std::vector<bool> locations = ball_detector->locate_balls();
153:
154:           if(!locations.at(0)) {  // move ball into top position
155:             upper_indexer->set_voltage(7500);
156:             lower_indexer->set_voltage(10500);
157:           } else if(locations.at(0) && !locations.at(1)) { // move ball from lowest/no position to middle position
158:             upper_indexer->set_voltage(0);
159:             lower_indexer->set_voltage(10500);
160:           } else { // indexer can't do anything to increment so don't run
161:             upper_indexer->set_voltage(0);
162:             lower_indexer->set_voltage(0);
163:           }
164:           break;
165:         } case e_fix_ball: {
166:           upper_indexer->set_voltage(-12000);
167:           pros::delay(250);
168:           upper_indexer->set_voltage(12000);
169:           pros::delay(500);
170:           upper_indexer->set_voltage(0);
171:           break;
172:         } case e_run_upper: {
173:           upper_indexer->set_voltage(12000);
174:           break;
175:         } case e_run_lower: {
176:           lower_indexer->set_voltage(12000);
177:           break;
178:         } case e_stop: {
179:           lower_indexer->set_voltage(0);
180:           upper_indexer->set_voltage(0);
181:           break;
182:         }
183:       }
184:
185:       if(action.command == e_index_until_filtered || action.command == e_index_to_state || action.command == e_fix_ball) {
186:         while ( command_finish_lock.exchange( true ) ); //aquire lock
187:         commands_finished.push_back(action.uid);
188:         command_finish_lock.exchange( false ); //release lock
189:       }
190:     }
191:   }
192:
193:   int Indexer::send_command(indexer_command command, indexer_args args /*{}*/) {
194:     while ( command_start_lock.exchange( true ) ); //aquire lock
195:     indexer_action action;
196:     action.command = command;
197:     action.args = args;
198:     action.uid = pros::millis() + lower_indexer->get_actual_voltage() + upper_indexer->get_actual_voltage();
199:     command_queue.push(action);
200:     command_start_lock.exchange( false ); //release lock
201:
202:     return action.uid;
203:   }
204:
205:   void Indexer::index() {
206:     send_command(e_index);
207:   }
208:
209:   void Indexer::filter() {
210:     send_command(e_filter);
211:   }
212:
213:   void Indexer::auto_index() {
214:     send_command(e_auto_index);
215:   }
216:
217:   void Indexer::index_no_backboard() {
218:     send_command(e_index_no_backboard);
219:   }
220:
221:   int Indexer::index_until_filtered(bool asynch /*false*/) {
222:     int uid = send_command(e_index_until_filtered);
223:
224:     if(!asynch) {
225:       wait_until_finished(uid);
226:     }
```

```
227:
228:     return uid;
229: }
230:
231: int Indexer::index_to_state(bool allow_filter, ball_positions end_state, bool asynch) {
232:     indexer_args args;
233:     args.allow_filter = allow_filter;
234:     args.end_state = end_state;
235:     int uid = send_command(e_index_to_state, args);
236:
237:     if(!asynch) {
238:         wait_until_finished(uid);
239:     }
240:
241:     return uid;
242: }
243:
244: void Indexer::increment() {
245:     send_command(e_increment);
246: }
247:
248: void Indexer::auto_increment() {
249:     send_command(e_auto_increment);
250: }
251:
252:
253:
254: void Indexer::run_upper_roller() {
255:     send_command(e_run_upper);
256: }
257:
258:
259: void Indexer::run_lower_roller() {
260:     send_command(e_run_lower);
261: }
262:
263:
264:
265: int Indexer::fix_ball(bool asynch /*true*/) {
266:     int uid = send_command(e_fix_ball);
267:
268:     if(!asynch) {
269:         wait_until_finished(uid);
270:     }
271:
272:     return uid;
273: }
274:
275:
276:
277: void Indexer::hard_stop() {
278:     reset_command_queue();
279:     send_command(e_stop);
280: }
281:
282: void Indexer::stop() {
283:     send_command(e_stop);
284: }
285:
286: ball_positions Indexer::get_state() {
287:     ball_positions state;
288:
289:     int color = ball_detector->check_filter_level();
290:     std::vector<bool> ball_locations = ball_detector->locate_balls();
291:     if(ball_locations.at(0)) {
292:         state.top = true;
293:     } else {
294:         state.top = false;
295:     }
296:     if(ball_locations.at(1)) {
297:         state.middle = true;
298:     } else {
299:         state.middle = false;
300:     }
301:
302:     if (color == 0) {
303:         state.middle_color = "none";
304:     } else if(color == 1) {
305:         state.middle_color = "blue";
306:     } else if (color == 2) {
307:         state.middle_color = "red";
308:     } else {
309:         state.middle_color = "unknown";
310:     }
311:
312:     return state;
313: }
314:
315: void Indexer::reset_command_queue() {
316:     while ( command_start_lock.exchange( true ) ); //aquire lock
317:     std::queue<indexer_action> empty_queue;
318:     std::swap( command_queue, empty_queue );  // replace command queue with an empty queue
319:     command_start_lock.exchange( false ); //release lock
320: }
321:
322:
323: void Indexer::update_filter_color(std::string new_color) {
324:     filter_color = new_color;
325: }
326:
327:
328: void Indexer::wait_until_finished(int uid) {
329:     while(std::find(commands_finished.begin(), commands_finished.end(), uid) == commands_finished.end()) {
330:         pros::delay(10);
331:     }
332:     while ( command_finish_lock.exchange( true ) ); //aquire lock
333:     commands_finished.erase(std::remove(commands_finished.begin(), commands_finished.end(), uid), commands_finished.end());
334:     command_finish_lock.exchange( false ); //release lock
335: }
336:
337:
338: bool Indexer::is_finished(int uid) {
339:     if(std::find(commands_finished.begin(), commands_finished.end(), uid) == commands_finished.end()) {
```

```
340:         while ( command_finish_lock.exchange( true ) ); //aquire lock
341:         commands_finished.erase(std::remove(commands_finished.begin(), commands_finished.end(), uid), commands_finished.end());
342:         command_finish_lock.exchange( false ); //release lock
343:
344:         return false;  // command is not finished because it is not in the list
345:       }
346:       return true;
347:   }
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/subsystems/chassis.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on: 2/16/2020
5:   * @reviewed_by: Aiden Carney
6:   *
7:   * Contains class for the chassis subsystem
8:   * has methods for driving during autonomous including turning and driving straight
9:   */
10:
11: #ifndef __CHASSIS_HPP__
12: #define __CHASSIS_HPP__
13:
14: #include <tuple>
15: #include <queue>
16:
17: #include "main.h"
18:
19: #include "../motors/Motor.hpp"
20: #include "../sensors/Sensors.hpp"
21:
22:
23: std::vector<double> generate_velocity_profile(int encoder_ticks, const std::function<double(double)>& max_acceleration, double max_decceleration, double max_velocity, double initial_velocity);
24:
25:
26: typedef enum {
27:     e_pid_straight_drive,
28:     e_profiled_straight_drive,
29:     e_turn,
30:     e_drive_to_point,
31:     e_turn_to_point,
32:     e_turn_to_angle
33: } chassis_commands;
34:
35: typedef struct {
36:     long double x;
37:     long double y;
38:     long double dx;
39:     long double dy;
40:     long double radius;
41:     long double dtheta;
42:     std::string get_string() {
43:         std::string str = (
44:             "{x: " + std::to_string(this->x)
45:             + " y: " + std::to_string(this->y)
46:             + " dx: " + std::to_string(this->dx)
47:             + " dy: " + std::to_string(this->dy)
48:             + " radius: " + std::to_string(this->radius)
49:             + " dtheta: " + std::to_string(this->dtheta)
50:             + "}"
51:         );
52:         return str;
53:     }
54: } waypoint;
55:
56: typedef struct {
57:     double setpoint1=0;
58:     double setpoint2=0;
59:     double kP=1;
60:     double kI=.001;
61:     double kD=.001;
62:     double I_max=INT32_MAX;
63:     int max_velocity=150;
64:     int timeout=INT32_MAX;
65:     int recalculations=0;
66:     int explicit_direction=0;
67:     double motor_slew=INT32_MAX;
68:     bool correct_heading=true;
69:     bool log_data=false;
70: } chassis_params;
71:
72: typedef struct {
73:     chassis_params args;
74:     int command_uid;
75:     chassis_commands command;
76: } chassis_action;
77:
78:
79: /**
80:  * @see: Motors.hpp
81:  *
82:  * contains methods to allow for easy control of the robot during
83:  * the autonomous period
84:  */
85: class Chassis
86: {
87:     private:
88:         static Motor *front_left_drive;
89:         static Motor *front_right_drive;
90:         static Motor *back_left_drive;
91:         static Motor *back_right_drive;
92:
93:         static Encoder* left_encoder;
94:         static Encoder* right_encoder;
95:
96:         pros::Task *thread;  // the motor thread
97:         static std::queue<chassis_action> command_queue;
98:         static std::vector<int> commands_finished;
99:         static std::atomic<bool> command_start_lock;
100:        static std::atomic<bool> command_finish_lock;
101:        static int num_instances;
102:
103:        static void t_pid_straight_drive(chassis_params args);  // functions called by thread for asynchronous movement
104:        static void t_profiled_straight_drive(chassis_params args);
105:        static void t_turn(chassis_params args);
106:        static void t_move_to_waypoint(chassis_params args, waypoint point);
107:
108:        static double wheel_diameter;
109:        static double width;
110:        static double gear_ratio;
111:
112:        static void chassis_motion_task(void*);
113:
```

```
114:
115:    public:
116:        Chassis( Motor &front_left, Motor &front_right, Motor &back_left, Motor &back_right, Encoder &l_encoder, Encoder &r_encoder, double chassis_width, double gearing=1, double wheel_size=4.05);
117:        ~Chassis();
118:
119:        int pid_straight_drive(double encoder_ticks, int relative_heading=0, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, bool correct_heading=true, double slew=0.2, bool log_data=true);
120:        int profiled_straight_drive(double encoder_ticks, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, bool correct_heading=true, int relative_heading=0, bool log_data=true);
121:        int uneven_drive(double l_enc_ticks, double r_enc_ticks, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, double slew=10, bool log_data=false);
122:        int turn_right(double degrees, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, double slew=15, bool log_data=true);
123:        int turn_left(double degrees, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, double slew=15, bool log_data=true);
124:        int drive_to_point(double x, double y, int recalculations=0, int explicit_direction=0, int max_velocity=450, int timeout=INT32_MAX, bool correct_heading=true, bool asynch=false, double slew=10, bool log_data=true);
125:        int turn_to_point(double x, double y, int max_velocity=450, int timeout=INT32_MAX, bool asynch = false, double slew=10, bool log_data=true);
126:        int turn_to_angle(double theta, int max_velocity=450, int timeout=INT32_MAX, bool asynch = false, double slew=10, bool log_data=true);
127:
128:        /**
129:         * @param: int voltage -> the voltage on interval [-127, 127] to set the motor to
130:         * @return: None
131:         *
132:         * sets voltage of chassis
133:         */
134:        void move( int voltage );
135:
136:        /**
137:         * @param: pros::motor_brake_mode_e_t new_brake_mode -> the new brakemode for the chassis
138:         * @return: None
139:         *
140:         * sets brake mode of all motors
141:         */
142:        void set_brake_mode( pros::motor_brake_mode_e_t new_brake_mode );
143:
144:
145:
146:        /**
147:         * @return: None
148:         *
149:         * @see: Motors.hpp
150:         *
151:         * changes the direction at the api motor level so that all the
152:         * motors in the chassis system are reversed
153:         * useful for allowing to change direction of drive in user control
154:         */
155:        void change_direction();
156:
157:        /**
158:         * @param: int speed -> the new speed the slew rate controller
159:         * @return: None
160:         *
161:         * sets the internal slew rate of the motor and enables it
162:         */
163:        void enable_slew( int rate=120 );
164:
165:        /**
166:         * @return: None
167:         *
168:         * disables internal slew rate of the motor
169:         */
170:        void disable_slew( );
171:
172:        void wait_until_finished(int uid);
173:        bool is_finished(int uid);
174:
175:
176:    };
177:
178:
179:    #endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/subsystems/chassis.cpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * @see: chassis.hpp
 *
 * contains implementation for chassis subsytem class
 */

#include <cmath>
#include <algorithm>
#include <deque>
#include <type_traits>

#include "main.h"

#include "../serial/Logger.hpp"
#include "../position_tracking/PositionTracker.hpp"
#include "chassis.hpp"


std::vector<double> generate_velocity_profile(int encoder_ticks, const std::function<double(double)>& max_acceleration, double max_decceleration, double max_velocity, double initial_velocity) {
    if(encoder_ticks <= 0) {
        Logger logger;
        log_entry entry;
        entry.content = (
            "[ERROR] " + std::string("PROFILE_CALCULATION")
            + ", Time: " + std::to_string(pros::millis())
            + ", Could not generate profile with negative or 0 encoder ticks"
            + ", enc_ticks: " + std::to_string(encoder_ticks)
            + ", max_velocity: " + std::to_string(max_velocity)
        );
        entry.stream = "clog";
        logger.add(entry);
        pros::delay(100); // add delay for msg to be logged
        throw std::invalid_argument("Cannot generate profile with negative or 0 encoder ticks");
    }

    std::vector<double> profile = {initial_velocity};

    int i = 0;
    while(i < encoder_ticks) {
        int ticks_left = encoder_ticks - i;
        int ticks_to_deccelerate = profile.at(i) / max_decceleration;
        if(ticks_to_deccelerate < ticks_left) {
            double step = profile.at(i) + max_acceleration(i);
            if(step > max_velocity) {
                step = max_velocity;
            }
            profile.push_back(step);
        } else {
            profile.push_back(profile.at(i) - max_decceleration);
        }

        i += 1;
    }

    return profile;
}


int Chassis::num_instances = 0;
std::queue<chassis_action> Chassis::command_queue;
std::vector<int> Chassis::commands_finished;
std::atomic<bool> Chassis::command_start_lock = ATOMIC_VAR_INIT(false);
std::atomic<bool> Chassis::command_finish_lock = ATOMIC_VAR_INIT(false);

Motor* Chassis::front_left_drive;
Motor* Chassis::front_right_drive;
Motor* Chassis::back_left_drive;
Motor* Chassis::back_right_drive;

Encoder* Chassis::left_encoder;
Encoder* Chassis::right_encoder;
double Chassis::width;
double Chassis::gear_ratio;
double Chassis::wheel_diameter;


Chassis::Chassis( Motor &front_left, Motor &front_right, Motor &back_left, Motor &back_right, Encoder &l_encoder, Encoder &r_encoder, double chassis_width, double gearing /*1*/, double wheel_size /*4.05*/)
{
    front_left_drive = &front_left;
    front_right_drive = &front_right;
    back_left_drive = &back_left;
    back_right_drive = &back_right;

    left_encoder = &l_encoder;
    right_encoder = &r_encoder;

    wheel_diameter = wheel_size;
    gear_ratio = gearing;
    width = chassis_width;

    if(num_instances == 0 || thread == NULL) {
        thread = new pros::Task( chassis_motion_task, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "chassis_thread");
    }

    num_instances += 1;

    front_left_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    front_right_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    back_left_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    back_right_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);

    front_left_drive->set_motor_mode(e_voltage);
    front_right_drive->set_motor_mode(e_voltage);
    back_left_drive->set_motor_mode(e_voltage);
    back_right_drive->set_motor_mode(e_voltage);

```

```
114:     front_left_drive->disable_slew();
115:     front_right_drive->disable_slew();
116:     back_left_drive->disable_slew();
117:     back_right_drive->disable_slew();
118:   }
119:
120:
121:
122:
123:   Chassis::~Chassis()
124:   {
125:     num_instances -= 1;
126:     if(num_instances == 0) {
127:       delete thread;
128:     }
129:   }
130:
131:
132:   // void Chassis::generate_profiles() {
133:   //    if(!profile_1.is_generated()) {
134:   //       profile_1.generate_profile(
135:   //          [](double n) -> double { return 183335300 + ((16.29262 - 183335300) / (1 + std::pow((n / 5807375), 1.381135))); },
136:   //          [](double n) -> double { return 201.3993 - (0.07738232 * n) - (0.0001796556 * std::pow(n, 2)); },
137:   //          // [](double n) -> double { return (201.3993 - (0.0967279 * n) - (0.0002807118 * std::pow(n, 2))); },
138:   //          // [](double n) -> double { return -14.87879 + ((199.9947 - -14.87879) / (1 + std::pow((n / 216.5658), 1.756448))); },  // 700 endpoint
139:   //          // [](double n) -> double { return -718.7411 + ((200.0491 - -718.7411) / (1 + std::pow((n / 3436.08), 0.8049193))); },  // 700 endpoint
140:   //          // [](double n) -> double { return -39.01548 + ((195.369 - -39.01548) / (1 + std::pow((n / 978.0565), 2.266577))); },  // 1700 endpoint
141:   //          // [](double n) -> double { return (-0.2 * n + 1000); },  // 1000 endpoint
142:   //          250,
143:   //          820,
144:   //          200,
145:   //          0
146:   //       );
147:   //    }
148:   // }
149:
150:
151:   void Chassis::chassis_motion_task(void*) {
152:     while(1) {
153:       if(command_queue.empty()) {  // delay unitl there is a command in the queue
154:         pros::delay(10);
155:         continue;
156:       }
157:
158:       // take lock and get command
159:       while ( command_start_lock.exchange( true ) ); //aquire lock
160:       chassis_action action = command_queue.front();
161:       command_queue.pop();
162:       command_start_lock.exchange( false ); //release lock
163:
164:       // execute command
165:       switch(action.command) {
166:         case e_pid_straight_drive:
167:           t_pid_straight_drive(action.args);
168:           break;
169:         case e_profiled_straight_drive:
170:           t_profiled_straight_drive(action.args);
171:           break;
172:         case e_turn:
173:           t_turn(action.args);
174:           break;
175:         case e_drive_to_point: {
176:           PositionTracker* tracker = PositionTracker::get_instance();
177:           std::vector<waypoint> waypoints;  // calculate waypoints based on starting position
178:
179:           long double dx = action.args.setpoint1 - tracker->get_position().x_pos;
180:           long double dy = action.args.setpoint2 - tracker->get_position().y_pos;
181:           std::cout << tracker->get_position().x_pos << " " << tracker->get_position().y_pos << "\n";
182:           // convert end coordinates to polar and then calculate waypoints
183:           long double delta_radius_polar = std::sqrt((std::pow(dx, 2) + std::pow(dy, 2)));
184:           long double delta_theta_polar = std::atan2(dy, dx);
185:
186:           for(int i=action.args.recalculations + 1; i > 0; i--) {  // calculate additional waypoints, start with last endpoint and go down
187:             long double radius = (i * delta_radius_polar) / (action.args.recalculations + 1);
188:             waypoint recalc_point;
189:             recalc_point.x = tracker->get_position().x_pos + (radius * std::cos(delta_theta_polar));  // intital x + dx
190:             recalc_point.y = tracker->get_position().y_pos + (radius * std::sin(delta_theta_polar));  // initial y + dy
191:             std::cout << radius << " " << delta_theta_polar << " " << (radius * std::cos(delta_theta_polar)) << " " << (radius * std::sin(delta_theta_polar)) << "\n";
192:             recalc_point.dx = radius * std::cos(delta_theta_polar);
193:             recalc_point.dy = radius * std::sin(delta_theta_polar);
194:             recalc_point.radius = radius;
195:             recalc_point.dtheta = delta_theta_polar;
196:             waypoints.insert(waypoints.begin(), recalc_point);
197:           }
198:           std::cout << "\n\n\n\n\n";
199:           if(action.args.log_data) {
200:             Logger logger;
201:             log_entry entry;
202:             std::string msg = (
203:               "[INFO] " + std::string("CHASSIS_ODOM")
204:               + ", Time: " + std::to_string(pros::millis())
205:               + ", dx: " + std::to_string(dx)
206:               + ", dy: " + std::to_string(dy)
207:               + ", delta_theta_polar: " + std::to_string(delta_theta_polar)
208:               + ", current x: " + std::to_string(tracker->get_position().x_pos)
209:               + ", current y: " + std::to_string(tracker->get_position().y_pos)
210:               + ", current theta: " + std::to_string(tracker->to_degrees(tracker->get_position().theta))
211:             );
212:             int i = 0;
213:             for(waypoint point : waypoints) {  // add waypoints to debug message
214:               msg += ", waypoint " + std::to_string(i) + ": " + point.get_string();
215:             }
216:             entry.content = msg;
217:             entry.stream = "clog";
218:             logger.add(entry);
219:           }
220:
221:           int start = pros::millis();
222:           for(waypoint point : waypoints) {  // move to each generated waypoint
223:             if(pros::millis() - start > action.args.timeout) {  // end early if past the timeout point
224:               break;
225:             }
226:             t_move_to_waypoint(action.args, point);
```

```
227:             }
228:             break;
229:         } case e_turn_to_point: {
230:             PositionTracker* tracker = PositionTracker::get_instance();
231:
232:             long double dx = action.args.setpoint1 - tracker->get_position().x_pos;
233:             long double dy = action.args.setpoint2 - tracker->get_position().y_pos;
234:
235:             // convert end coordinates to polar to find the change in angle
236:             // long double dtheta = std::fmod((-M_PI / 2) + std::atan2(dy, dx), (2 * M_PI));
237:             long double dtheta = std::atan2(dy, dx);
238:             if(dtheta < 0) { // map to [0, 2pi]
239:                 dtheta += 2 * M_PI;
240:             }
241:
242:             // current angle is bounded by [-pi, pi] re map it to [0, 2pi]
243:             long double current_angle = tracker->get_heading_rad();
244:             if(current_angle < 0) {
245:                 current_angle += 2 * M_PI;
246:             }
247:             current_angle = (-current_angle) + (M_PI / 2);
248:
249:             // calculate how much the robot needs to turn to be at the angle
250:             long double to_turn_face_forwards = current_angle - dtheta; // change in robot angle
251:             long double to_turn_face_backwards = (current_angle - dtheta) - M_PI;
252:
253:             if(to_turn_face_forwards > M_PI) { // find minimal angle change and direction of change [-PI/2, PI/2]
254:                 to_turn_face_forwards = (-2 * M_PI) + to_turn_face_forwards; // give negative value to turn left to point
255:             } else if(to_turn_face_forwards < -M_PI) {
256:                 to_turn_face_forwards = (2 * M_PI) + to_turn_face_forwards; // give positive value to turn left to point
257:             }
258:
259:             if(to_turn_face_backwards > M_PI) { // find minimal angle change and direction of change [-PI/2, PI/2]
260:                 to_turn_face_backwards = (-2 * M_PI) + to_turn_face_backwards; // give negative value to turn left to point
261:             } else if(to_turn_face_backwards < -M_PI) {
262:                 to_turn_face_backwards = (2 * M_PI) + to_turn_face_backwards; // give positive value to turn left to point
263:             }
264:
265:
266:             long double to_turn;
267:             int direction;
268:             if(action.args.explicit_direction == 1) { // force positive direction
269:                 to_turn = to_turn_face_forwards;
270:                 direction = 1;
271:             } else if(action.args.explicit_direction == -1) { // force negative direction
272:                 to_turn = to_turn_face_backwards;
273:                 direction = -1;
274:             } else if(std::abs(to_turn_face_forwards) < std::abs(to_turn_face_backwards)) { // faster to go forwards
275:                 to_turn = to_turn_face_forwards;
276:                 direction = 1;
277:             } else { // faster to go backwards
278:                 to_turn = to_turn_face_backwards;
279:                 direction = -1;
280:             }
281:
282:             to_turn = tracker->to_degrees(to_turn);
283:
284:
285:             // set up turn
286:             chassis_params turn_args;
287:             turn_args.setpoint1 = to_turn;
288:             turn_args.max_velocity = action.args.max_velocity;
289:             turn_args.timeout = 15000; // TODO: add time estimation
290:             turn_args.kP = 2.8;
291:             turn_args.kI = 0.0005;
292:             turn_args.kD = 50;
293:             turn_args.I_max = INT32_MAX;
294:             turn_args.motor_slew = action.args.motor_slew;
295:             turn_args.log_data = action.args.log_data;
296:
297:             // perform turn
298:             std::cout << "starting turn\n";
299:             std::cout << to_turn << "\n";
300:             t_turn(turn_args);
301:             std::cout << "turn done\n";
302:
303:             if(action.args.log_data) {
304:                 Logger logger;
305:                 log_entry entry;
306:                 entry.content = (
307:                     "[INFO] " + std::string("CHASSIS_ODOM")
308:                     + ", Time: " + std::to_string(pros::millis())
309:                     + ", X " + std::to_string(action.args.setpoint1)
310:                     + ", Y " + std::to_string(action.args.setpoint2)
311:                     + ", ToTurnForwards: " + std::to_string(tracker->to_degrees(to_turn_face_forwards))
312:                     + ", ToTurnBackwards: " + std::to_string(tracker->to_degrees(to_turn_face_backwards))
313:                     + ", ToTurn: " + std::to_string(to_turn)
314:                     + ", Direction: " + std::to_string(direction)
315:                     + ", dx: " + std::to_string(dx)
316:                     + ", dy: " + std::to_string(dy)
317:                     + ", X: " + std::to_string(tracker->get_position().x_pos)
318:                     + ", Y: " + std::to_string(tracker->get_position().y_pos)
319:                     + ", Theta: " + std::to_string(tracker->to_degrees(tracker->get_position().theta))
320:                 );
321:                 entry.stream = "clog";
322:                 logger.add(entry);
323:             }
324:
325:             pros::delay(100); // add delay for extra settling
326:
327:             break;
328:         } case e_turn_to_angle: {
329:             PositionTracker* tracker = PositionTracker::get_instance();
330:
331:             // current angle is bounded by [-pi, pi] re map it to [0, pi]
332:             long double current_angle = tracker->get_heading_rad();
333:             if(current_angle < -M_PI) {
334:                 current_angle += M_PI;
335:             }
336:
337:             // calculate how much the robot needs to turn to be at the angle
338:             long double to_turn = action.args.setpoint1 - current_angle; // change in robot angle
339:
```

```
340:             if(to_turn > M_PI) {  // find minimal angle change and direction of change [-PI/2, PI/2]
341:                 to_turn = (-2 * M_PI) + to_turn;  // give negative value to turn left to point
342:             } else if(to_turn < -M_PI) {
343:                 to_turn = (2 * M_PI) + to_turn;  // give positive value to turn left to point
344:             }
345:             to_turn = tracker->to_degrees(to_turn);
346:
347:             std::cout << current_angle << " " << to_turn << "\n";
348:
349:             // set up turn
350:             chassis_params turn_args;
351:             turn_args.setpoint1 = to_turn;
352:             turn_args.max_velocity = action.args.max_velocity;
353:             turn_args.timeout = action.args.timeout; // TODO: add time estimation
354:             turn_args.kP = 2.8;
355:             turn_args.kI = 0.0005;
356:             turn_args.kD = 50;
357:             turn_args.I_max = INT32_MAX;
358:             turn_args.motor_slew = action.args.motor_slew;
359:             turn_args.log_data = action.args.log_data;
360:
361:             if(action.args.log_data) {
362:                 Logger logger;
363:                 log_entry entry;
364:                 std::string msg = (
365:                     "[INFO] " + std::string("CHASSIS_ODOM")
366:                     + ", Time: " + std::to_string(pros::millis())
367:                     + ", turning: " + std::to_string(to_turn)
368:                     + ", Current re-bounded angle: " + std::to_string(tracker->to_degrees(current_angle))
369:                     + ", Current angle: " + std::to_string(tracker->to_degrees(tracker->get_heading_rad()))
370:                 );
371:                 entry.content = msg;
372:                 entry.stream = "clog";
373:                 logger.add(entry);
374:             }
375:
376:             // perform turn
377:             t_turn(turn_args);
378:
379:             break;
380:         }
381:     }
382:
383:     while ( command_finish_lock.exchange( true ) ); //aquire lock
384:     commands_finished.push_back(action.command_uid);
385:     command_finish_lock.exchange( false ); //release lock
386: }
387: }
388:
389:
390:
391:
392: void Chassis::t_pid_straight_drive(chassis_params args) {
393:     PositionTracker* tracker = PositionTracker::get_instance();
394:     Configuration* config = Configuration::get_instance();
395:
396:     double kP_l = args.kP;
397:     double kI_l = args.kI;
398:     double kD_l = args.kD;
399:     double I_max_l = args.I_max;
400:
401:     double kP_r = kP_l;
402:     double kI_r = kI_l;
403:     double kD_r = kD_l;
404:     double I_max_r = I_max_l;
405:
406:     front_left_drive->disable_driver_control();
407:     front_right_drive->disable_driver_control();
408:     back_left_drive->disable_driver_control();
409:     back_right_drive->disable_driver_control();
410:
411:     front_left_drive->set_motor_mode(e_builtin_velocity_pid);
412:     front_right_drive->set_motor_mode(e_builtin_velocity_pid);
413:     back_left_drive->set_motor_mode(e_builtin_velocity_pid);
414:     back_right_drive->set_motor_mode(e_builtin_velocity_pid);
415:
416:     int r_id = right_encoder->get_unique_id(true);
417:     int l_id = left_encoder->get_unique_id(true);
418:
419:     double integral_l = 0;
420:     double integral_r = 0;
421:     double prev_error_l = 0;
422:     double prev_error_r = 0;
423:     double prev_velocity_l = 0;
424:     double prev_velocity_r = 0;
425:
426:     double prev_l_encoder = std::get<0>(Sensors::get_average_encoders(l_id, r_id));
427:     double prev_r_encoder = std::get<1>(Sensors::get_average_encoders(l_id, r_id));
428:
429:     long double relative_angle = 0;
430:     long double abs_angle = tracker->to_degrees(tracker->get_heading_rad());
431:     long double prev_abs_angle = abs_angle;
432:     long double integral_heading = 0;
433:     double prev_heading_error = 0;
434:
435:     bool settled = false;
436:     std::vector<double> previous_l_velocities;
437:     std::vector<double> previous_r_velocities;
438:     int velocity_history = 15;
439:     bool use_integral_l = true;
440:     bool use_integral_r = true;
441:
442:     int current_time = pros::millis();
443:     int start_time = current_time;
444:
445:     do {
446:         int dt = pros::millis() - current_time;
447:         // pid distance controller
448:         double error_l = args.setpoint1 - std::get<0>(Sensors::get_average_encoders(l_id, r_id));
449:         double error_r = args.setpoint2 - std::get<1>(Sensors::get_average_encoders(l_id, r_id));
450:
451:         if ( std::abs(integral_l) > I_max_l || !use_integral_l) {
452:             integral_l = 0;  // reset integral if greater than max allowable value
```

```
453:            use_integral_l = false;
454:        } else {
455:            integral_l = integral_l + (error_l * dt);
456:        }
457:
458:        if ( std::abs(integral_r) > I_max_l || !use_integral_r) {
459:            integral_r = 0;  // reset integral if greater than max allowable value
460:            use_integral_r = false;
461:        } else {
462:            integral_r = integral_r + (error_r * dt);
463:        }
464:
465:        current_time = pros::millis();
466:
467:        double derivative_l = error_l - prev_error_l;
468:        double derivative_r = error_r - prev_error_r;
469:        prev_error_l = error_l;
470:        prev_error_r = error_r;
471:
472:        double left_velocity = (kP_l * error_l) + (kI_l * integral_l) + (kD_l * derivative_l);
473:        double right_velocity = (kP_r * error_r) + (kI_r * integral_r) + (kD_r * derivative_r);
474:
475:
476:    // slew rate code
477:        double delta_velocity_l = left_velocity - prev_velocity_l;
478:        double delta_velocity_r = right_velocity - prev_velocity_r;
479:        double slew_rate = args.motor_slew;
480:        if(std::abs(delta_velocity_l) > (dt * slew_rate) && (std::signbit(delta_velocity_l) == std::signbit(left_velocity)) ) {  // ignore deceleration
481:            int sign = std::abs(delta_velocity_l) / delta_velocity_l;
482:            std::cout << "l over slew: " << sign << " " << dt << " " << slew_rate << "\n";
483:            left_velocity = prev_velocity_l + (sign * dt * slew_rate);
484:        }
485:
486:        if(std::abs(delta_velocity_r) > (dt * slew_rate) && (std::signbit(delta_velocity_r) == std::signbit(right_velocity))) {
487:            int sign = std::abs(delta_velocity_r) / delta_velocity_r;
488:            std::cout << "r over slew: " << sign << " " << dt << " " << slew_rate << "\n";
489:            right_velocity = prev_velocity_r + (sign * dt * slew_rate);
490:        }
491:
492:
493:    // p controller heading correction
494:        abs_angle = tracker->get_heading_rad();
495:        abs_angle = std::atan2(std::sin(abs_angle), std::cos(abs_angle));
496:        long double delta_theta;
497:        // account for angle wrap around    ie. new = -1, prev = -359   == bad delta
498:        if(prev_abs_angle > 0 && abs_angle < 0 && std::abs(tracker->to_radians(prev_abs_angle)) + std::abs(abs_angle) > (M_PI)) {
499:            delta_theta = tracker->to_degrees((2*M_PI) + abs_angle) - prev_abs_angle;
500:        } else if(prev_abs_angle < 0 && abs_angle > 0 && std::abs(tracker->to_radians(prev_abs_angle)) + std::abs(abs_angle) > (M_PI)) {
501:            delta_theta = tracker->to_degrees(abs_angle - (2*M_PI)) - prev_abs_angle;
502:        } else {
503:            delta_theta = tracker->to_degrees(abs_angle) - prev_abs_angle;
504:        }
505:
506:        relative_angle += delta_theta;
507:        prev_abs_angle = tracker->to_degrees(abs_angle);
508:
509:
510:        double heading_error = 0 - relative_angle;  // 0 is the setpoint because we want to drive straight
511:        integral_heading = integral_heading + (heading_error * dt);
512:        double d_heading_error = heading_error - prev_heading_error;
513:        prev_heading_error = heading_error;
514:        // std::cout << "delta_theta: " << delta_theta << " | prev_anlge: " << prev_angle << " | relative angle: " << relative_angle << " | heading_error: " << heading_error << "\n";
515:        int velocity_correction = (.05 * heading_error) + (0 * integral_heading) + (0 * d_heading_error);
516:        // int velocity_correction = (4 * heading_error) + (0 * integral_heading) + (54 * d_heading_error);
517:        if(args.correct_heading && heading_error > 0.00001) {  // veering left
518:            // velocity_correction = 5;
519:            right_velocity -= velocity_correction;
520:        } else if ( args.correct_heading && heading_error < -0.00001) {  // veering right
521:            // velocity_correction = 5;
522:            left_velocity -= velocity_correction;
523:        }
524:
525:
526:        // cap voltage to max voltage with regard to velocity
527:        if ( std::abs(left_velocity) > args.max_velocity ) {
528:            left_velocity = left_velocity > 0 ? args.max_velocity : -args.max_velocity;
529:        }
530:        if ( std::abs(right_velocity) > args.max_velocity ) {
531:            right_velocity = right_velocity > 0 ? args.max_velocity : -args.max_velocity;
532:        }
533:
534:        if ( args.log_data ) {
535:            Logger logger;
536:            log_entry entry;
537:            entry.content = (
538:                "[INFO] " + std::string("CHASSIS_PID")
539:                + ", Time: " + std::to_string(pros::millis())
540:                + ", Actual_Vol1: " + std::to_string(front_left_drive->get_actual_voltage())
541:                + ", Actual_Vol2: " + std::to_string(front_right_drive->get_actual_voltage())
542:                + ", Actual_Vol3: " + std::to_string(back_left_drive->get_actual_voltage())
543:                + ", Actual_Vol4: " + std::to_string(back_right_drive->get_actual_voltage())
544:                + ", Slew: " + std::to_string(args.motor_slew)
545:                + ", Brake: " + std::to_string(front_left_drive->get_brake_mode())
546:                + ", Gear: " + std::to_string(front_left_drive->get_gearset())
547:                + ", I_max: " + std::to_string(I_max_l)
548:                + ", I: " + std::to_string(integral_l)
549:                + ", kD: " + std::to_string(kD_l)
550:                + ", kI: " + std::to_string(kI_l)
551:                + ", kP: " + std::to_string(kP_l)
552:                + ", Position_Sp: " + std::to_string(args.setpoint1)
553:                + ", position_l: " + std::to_string(std::get<0>(Sensors::get_average_encoders(l_id, r_id)))
554:                + ", position_r: " + std::to_string(std::get<1>(Sensors::get_average_encoders(l_id, r_id)))
555:                + ", Heading_Sp: " + std::to_string(args.setpoint2)
556:                + ", Relative_Heading: " + std::to_string(relative_angle)
557:                + ", Actual_Vel1: " + std::to_string(front_left_drive->get_actual_velocity())
558:                + ", Actual_Vel2: " + std::to_string(front_right_drive->get_actual_velocity())
559:                + ", Actual_Vel3: " + std::to_string(back_left_drive->get_actual_velocity())
560:                + ", Actual_Vel4: " + std::to_string(back_right_drive->get_actual_velocity())
561:            );
562:            entry.stream = "clog";
563:            logger.add(entry);
564:        }
565:
```

```
566:        prev_velocity_l = left_velocity;
567:        prev_velocity_r = right_velocity;
568:
569:        previous_l_velocities.push_back(left_velocity);
570:        previous_r_velocities.push_back(right_velocity);
571:        if(previous_l_velocities.size() > velocity_history) {
572:            previous_l_velocities.erase(previous_l_velocities.begin());
573:        }
574:
575:        if(previous_r_velocities.size() > velocity_history) {
576:            previous_r_velocities.erase(previous_r_velocities.begin());
577:        }
578:
579:        // settled is when error is almost zero and velocity is minimal
580:        double l_difference = *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second - *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).first;
581:        double r_difference = *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).second - *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).first;
582:        std::cout << "difference: " << *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second << " " << previous_l_velocities.size() << "\n";
583:        if (
584:            std::abs(l_difference) < 2
585:            && previous_l_velocities.size() == velocity_history
586:            && std::abs(r_difference) < 2
587:            && previous_r_velocities.size() == velocity_history
588:            && left_velocity < 2
589:            && right_velocity < 2
590:        ) {
591:            break; // end before timeout
592:        }
593:
594:
595:        front_left_drive->move_velocity(left_velocity);
596:        front_right_drive->move_velocity(right_velocity);
597:        back_left_drive->move_velocity(left_velocity);
598:        back_right_drive->move_velocity(right_velocity);
599:
600:        pros::delay(10);
601:    } while ( pros::millis() < start_time + args.timeout );
602:
603:    front_left_drive->set_motor_mode(e_voltage);
604:    front_right_drive->set_motor_mode(e_voltage);
605:    back_left_drive->set_motor_mode(e_voltage);
606:    back_right_drive->set_motor_mode(e_voltage);
607:
608:    front_left_drive->set_voltage(0);
609:    front_right_drive->set_voltage(0);
610:    back_left_drive->set_voltage(0);
611:    back_right_drive->set_voltage(0);
612:
613:    front_left_drive->enable_driver_control();
614:    front_right_drive->enable_driver_control();
615:    back_left_drive->enable_driver_control();
616:    back_right_drive->enable_driver_control();
617:
618:    right_encoder->forget_position(r_id);  // free up space in the encoders log
619:    left_encoder->forget_position(l_id);
620: }
621:
622:
623:
624:
625: void Chassis::t_profiled_straight_drive(chassis_params args) {
626:    PositionTracker* tracker = PositionTracker::get_instance();
627:    Configuration* config = Configuration::get_instance();
628:
629:    double kP = args.kP;
630:    double kI = args.kI;
631:    double kD = args.kD;
632:    double I_max = INT32_MAX;
633:
634:    front_left_drive->disable_driver_control();
635:    front_right_drive->disable_driver_control();
636:    back_left_drive->disable_driver_control();
637:    back_right_drive->disable_driver_control();
638:
639:    front_left_drive->set_motor_mode(e_builtin_velocity_pid);
640:    front_right_drive->set_motor_mode(e_builtin_velocity_pid);
641:    back_left_drive->set_motor_mode(e_builtin_velocity_pid);
642:    back_right_drive->set_motor_mode(e_builtin_velocity_pid);
643:
644:    int r_id = right_encoder->get_unique_id(true);
645:    int l_id = left_encoder->get_unique_id(true);
646:
647:    long double relative_angle = 0;
648:    long double abs_angle = tracker->to_degrees(tracker->get_heading_rad());
649:    long double prev_abs_angle = abs_angle;
650:    long double integral = 0;
651:    long double prev_integral = 0;
652:    double prev_error = 0;
653:    bool use_integral = true;
654:
655:    int current_time = pros::millis();
656:    int start_time = current_time;
657:    bool settled = false;
658:    bool was_at_target_l = false;
659:    bool was_at_target_r = false;
660:
661:    std::vector<double> previous_l_velocities;
662:    std::vector<double> previous_r_velocities;
663:    int velocity_history = 15;
664:
665:    auto accel_func = [](double n) -> double { return 0.005 * n; };
666:    // auto accel_func = [](double n) -> double { return 1; };
667:    std::vector<double> velocity_profile = generate_velocity_profile(std::abs(args.setpoint1), accel_func, .55, args.max_velocity, 50);  // .45 is decceleration, 10 is initial velocity
668:
669:    do {
670:        int dt = pros::millis() - current_time;
671:        current_time = pros::millis();
672:
673:        double velocity_l;
674:        double velocity_r;
675:        if(std::abs(std::get<0>(Sensors::get_average_encoders(l_id, r_id))) <= std::abs(args.setpoint1)) {
676:            velocity_l = velocity_profile.at(std::abs(std::get<0>(Sensors::get_average_encoders(l_id, r_id))));
677:        } else {
678:            was_at_target_l = true;
```

```
679:            velocity_l = 0;
680:        }
681:
682:        if(std::abs(std::get<1>(Sensors::get_average_encoders(l_id, r_id))) <= std::abs(args.setpoint1)) {
683:            velocity_r = velocity_profile.at(std::abs(std::get<1>(Sensors::get_average_encoders(l_id, r_id))));
684:        } else {
685:            was_at_target_l = true;
686:            velocity_r = 0;
687:        }
688:
689:
690:        // double velocity;
691:        // if(velocity_l > velocity_r) {
692:        //     velocity_r = velocity_r;
693:        //     velocity_l = velocity_r;
694:        // } else {
695:        //     velocity_r = velocity_l;
696:        //     velocity_l = velocity_l;
697:        // }
698:
699:        if(args.setpoint1 < 0) {
700:            velocity_l = -velocity_l;
701:            velocity_r = -velocity_r;
702:        }
703:
704:        abs_angle = tracker->to_degrees(tracker->get_heading_rad());
705:        long double delta_theta = abs_angle - prev_abs_angle;
706:        relative_angle += delta_theta;
707:        prev_abs_angle = abs_angle;
708:
709:        long double error = 0 - relative_angle;  // setpoint is 0 because we want to drive straight
710:        // long double error = std::get<0>(Sensors::get_average_encoders(l_id, r_id)) - std::get<1>(Sensors::get_average_encoders(l_id, r_id));
711:        std::cout << "relative angle: " << relative_angle << " | dtheta: " << delta_theta << "\n";
712:        // cap velocity to max velocity with regard to velocity
713:        integral = integral + (error * dt);
714:        if(integral > I_max) {
715:            integral = I_max;
716:        } else if (integral < -I_max) {
717:            integral = -I_max;
718:        }
719:
720:        // if(std::signbit(error) != std::signbit(prev_error)) {
721:        //     std::cout << "halving " << integral << " " << prev_integral;
722:        //     integral = .5 * integral;
723:        // }
724:        prev_integral = integral;
725:
726:        double derivative = error - prev_error;
727:        prev_error = error;
728:
729:
730:        // std::cout << error << " " << relative_angle << "\n";
731:
732:    // pid heading correction
733:        // int velocity_correction = (kP * error) + (kI * integral) + (kD * derivative);
734:
735:    // PI heading correction
736:        // double velocity_correction = std::abs(kI * integral);
737:        double velocity_correction = std::abs(args.kP * error + args.kI * integral + args.kD * derivative);
738:        std::cout << "integral: " << integral << " " << velocity_correction << "\n";
739:        if(args.correct_heading  && error > 0.000001) {  // veering off course, so correct
740:            velocity_r -= velocity_correction / 2;
741:            velocity_l += velocity_correction / 2;
742:        } else if(args.correct_heading && error < -0.000001) {
743:            velocity_l -= velocity_correction / 2;
744:            velocity_r += velocity_correction / 2;
745:        }
746:
747:
748:    // cap velocity to max velocity with regard to direction
749:        if ( std::abs(velocity_l) > args.max_velocity ) {
750:            velocity_l = velocity_l > 0 ? args.max_velocity : -args.max_velocity;
751:        }
752:        if ( std::abs(velocity_r) > args.max_velocity ) {
753:            velocity_r = velocity_r > 0 ? args.max_velocity : -args.max_velocity;
754:        }
755:
756:        if ( args.log_data ) {
757:            Logger logger;
758:            log_entry entry;
759:            entry.content = (
760:                "[INFO] " + std::string("CHASSIS_PROFILED_STRAIGHT_DRIVE")
761:                + ", Time: " + std::to_string(pros::millis())
762:                + ", Actual_Vol1: " + std::to_string(front_left_drive->get_actual_voltage())
763:                + ", Actual_Vol2: " + std::to_string(front_right_drive->get_actual_voltage())
764:                + ", Actual_Vol3: " + std::to_string(back_left_drive->get_actual_voltage())
765:                + ", Actual_Vol4: " + std::to_string(back_right_drive->get_actual_voltage())
766:                + ", Slew: " + std::to_string(args.motor_slew)
767:                + ", Brake: " + std::to_string(front_left_drive->get_brake_mode())
768:                + ", Gear: " + std::to_string(front_left_drive->get_gearset())
769:                + ", I_max: " + std::to_string(I_max)
770:                + ", I: " + std::to_string(integral)
771:                + ", kD: " + std::to_string(kD)
772:                + ", kI: " + std::to_string(kI)
773:                + ", kP: " + std::to_string(kP)
774:                + ", Position_Sp: " + std::to_string(args.setpoint1)
775:                + ", position_l: " + std::to_string(std::get<0>(Sensors::get_average_encoders(l_id, r_id)))
776:                + ", position_r: " + std::to_string(std::get<1>(Sensors::get_average_encoders(l_id, r_id)))
777:                + ", Heading_Sp: " + std::to_string(args.setpoint2)
778:                + ", Relative_Heading: " + std::to_string(relative_angle)
779:                + ", Actual_Vel1: " + std::to_string(velocity_l)
780:                + ", Actual_Vel2: " + std::to_string(velocity_r)
781:                + ", Actual_Vel3: " + std::to_string(back_left_drive->get_actual_velocity())
782:                + ", Actual_Vel4: " + std::to_string(back_right_drive->get_actual_velocity())
783:                + ", Correction: " + std::to_string(velocity_correction)
784:            );
785:            entry.stream = "clog";
786:            logger.add(entry);
787:        }
788:
789:        double error_l = std::abs(args.setpoint1 - std::get<0>(Sensors::get_average_encoders(l_id, r_id)));
790:        double error_r = std::abs(args.setpoint1 - std::get<1>(Sensors::get_average_encoders(l_id, r_id)));
791:
```

```
792:         previous_l_velocities.push_back(velocity_l);
793:         previous_r_velocities.push_back(velocity_r);
794:         if(previous_l_velocities.size() > velocity_history) {
795:             previous_l_velocities.erase(previous_l_velocities.begin());
796:         }
797:         if(previous_r_velocities.size() > velocity_history) {
798:             previous_r_velocities.erase(previous_r_velocities.begin());
799:         }
800:
801:         // settled is when error is almost zero and velocity is minimal
802:         double l_difference = *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second - *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).first;
803:         double r_difference = *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).second - *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).first;
804:         if (
805:             std::abs(l_difference) < 2
806:             && previous_l_velocities.size() == velocity_history
807:             && std::abs(r_difference) < 2
808:             && previous_r_velocities.size() == velocity_history
809:             && std::abs(velocity_l) < 2
810:             && std::abs(velocity_r) < 2
811:         ) {
812:             break; // end before timeout
813:         }
814:         // if(error_l < 5 || error_r < 5) {  // shut off motors when one side reaches the setpoint
815:         //     velocity_l = 0;
816:         //     velocity_r = 0;
817:         //     break;
818:         // }
819:
820:         std::cout << "velocity: " << velocity_l << " " << velocity_r << "\n";
821:         std::cout << "error: " << error_r << " " << error_l << " " << error << "\n";
822:         front_left_drive->move_velocity(velocity_l);
823:         front_right_drive->move_velocity(velocity_r);
824:         back_left_drive->move_velocity(velocity_l);
825:         back_right_drive->move_velocity(velocity_r);
826:
827:         pros::delay(10);
828:     } while (pros::millis() < start_time + args.timeout);
829:
830:     front_left_drive->set_motor_mode(e_voltage);
831:     front_right_drive->set_motor_mode(e_voltage);
832:     back_left_drive->set_motor_mode(e_voltage);
833:     back_right_drive->set_motor_mode(e_voltage);
834:
835:     front_left_drive->set_voltage(0);
836:     front_right_drive->set_voltage(0);
837:     back_left_drive->set_voltage(0);
838:     back_right_drive->set_voltage(0);
839:
840:     front_left_drive->enable_driver_control();
841:     front_right_drive->enable_driver_control();
842:     back_left_drive->enable_driver_control();
843:     back_right_drive->enable_driver_control();
844:
845:     front_left_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
846:     front_right_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
847:     back_left_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
848:     back_right_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
849:
850:     right_encoder->forget_position(r_id);  // free up space in the encoders log
851:     left_encoder->forget_position(l_id);
852: }
853:
854:
855:
856:
857: void Chassis::t_turn(chassis_params args) {
858:     PositionTracker* tracker = PositionTracker::get_instance();
859:     Configuration* config = Configuration::get_instance();
860:
861:     double kP = args.kP;
862:     double kI = args.kI;
863:     double kD = args.kD;
864:     double I_max = args.I_max;
865:
866:     front_left_drive->disable_driver_control();
867:     front_right_drive->disable_driver_control();
868:     back_left_drive->disable_driver_control();
869:     back_right_drive->disable_driver_control();
870:
871:     front_left_drive->set_motor_mode(e_builtin_velocity_pid);
872:     front_right_drive->set_motor_mode(e_builtin_velocity_pid);
873:     back_left_drive->set_motor_mode(e_builtin_velocity_pid);
874:     back_right_drive->set_motor_mode(e_builtin_velocity_pid);
875:
876:     front_left_drive->move_velocity(0);
877:     front_right_drive->move_velocity(0);
878:     back_left_drive->move_velocity(0);
879:     back_right_drive->move_velocity(0);
880:
881:     front_left_drive->tare_encoder();
882:     front_right_drive->tare_encoder();
883:     back_left_drive->tare_encoder();
884:     back_right_drive->tare_encoder();
885:
886:     int r_id = right_encoder->get_unique_id();
887:     int l_id = left_encoder->get_unique_id();
888:     right_encoder->reset(r_id);
889:     left_encoder->reset(l_id);
890:
891:     long double relative_angle = 0;
892:     long double abs_angle = tracker->to_degrees(tracker->get_heading_rad());
893:     long double prev_abs_angle = abs_angle;
894:     long double integral = 0;
895:     double prev_error = 0;
896:     bool use_integral = true;
897:     int current_time = pros::millis();
898:     int start_time = current_time;
899:
900:     double prev_velocity_l = 0;
901:     double prev_velocity_r = 0;
902:
903:     // std::vector<double> previous_l_velocities;
904:     // std::vector<double> previous_r_velocities;
```

```
905:    // int velocity_history = 15;
906:    std::vector<double> error_history;
907:    int max_history_length = 20;
908:
909:    do {
910:        int dt = pros::millis() - current_time;
911:
912:        abs_angle = tracker->get_heading_rad();
913:        abs_angle = std::atan2(std::sin(abs_angle), std::cos(abs_angle));
914:        long double delta_theta;
915:        // account for angle wrap around
916:        // ie. new = -1, prev = -359   == bad delta
917:        if(prev_abs_angle > 0 && abs_angle < 0 && std::abs(tracker->to_radians(prev_abs_angle)) + std::abs(abs_angle) > (M_PI)) {
918:            delta_theta = tracker->to_degrees((2*M_PI) + abs_angle) - prev_abs_angle;
919:        } else if(prev_abs_angle < 0 && abs_angle > 0 && std::abs(tracker->to_radians(prev_abs_angle)) + std::abs(abs_angle) > (M_PI)) {
920:            delta_theta = tracker->to_degrees(abs_angle - (2*M_PI)) - prev_abs_angle;
921:        } else {
922:            delta_theta = tracker->to_degrees(abs_angle) - prev_abs_angle;
923:        }
924:        // long double delta_theta = abs_angle - prev_abs_angle;
925:        // long double delta_theta = tracker->to_degrees(tracker->get_delta_theta_rad());
926:        relative_angle += delta_theta;
927:        prev_abs_angle = tracker->to_degrees(abs_angle);
928:
929:        long double error = args.setpoint1 - relative_angle;
930:
931:        integral = integral + (error * dt);
932:        if(integral > I_max) {
933:            integral = I_max;
934:        } else if (integral < -I_max) {
935:            integral = -I_max;
936:        }
937:
938:        double derivative = error - prev_error;
939:        prev_error = error;
940:
941:        current_time = pros::millis();
942:
943:        // std::cout << "relative angle: " << relative_angle << " | dtheta: " << delta_theta << "\n";
944:        // std::cout << error << " " << relative_angle << "\n";
945:
946:        double abs_velocity = (kP * error) + (kI * integral) + (kD * derivative);
947:        double l_velocity = abs_velocity;
948:        double r_velocity = -abs_velocity;
949:
950:        // slew rate code
951:        double delta_velocity_l = l_velocity - prev_velocity_l;
952:        double delta_velocity_r = r_velocity - prev_velocity_r;
953:        double slew_rate = args.motor_slew;
954:        int over_slew = 0;
955:        if(std::abs(delta_velocity_l) > (dt * slew_rate) && (std::signbit(delta_velocity_l) == std::signbit(l_velocity)) ) {  // ignore deceleration
956:            int sign = std::abs(delta_velocity_l) / delta_velocity_l;
957:            std::cout << "l over slew: " << sign << " " << dt << " " << slew_rate << "\n";
958:            l_velocity = prev_velocity_l + (sign * dt * slew_rate);
959:            over_slew = 1;
960:        }
961:
962:        if(std::abs(delta_velocity_r) > (dt * slew_rate) && (std::signbit(delta_velocity_r) == std::signbit(r_velocity))) {
963:            int sign = std::abs(delta_velocity_r) / delta_velocity_r;
964:            std::cout << "r over slew: " << sign << " " << dt << " " << slew_rate << "\n";
965:            r_velocity = prev_velocity_r + (sign * dt * slew_rate);
966:            over_slew = 1;
967:        }
968:
969:
970:        // cap velocity to max velocity with regard to velocity
971:        if ( std::abs(l_velocity) > args.max_velocity ) {
972:            l_velocity = l_velocity > 0 ? args.max_velocity : -args.max_velocity;
973:        }
974:        if ( std::abs(r_velocity) > args.max_velocity ) {
975:            r_velocity = r_velocity > 0 ? args.max_velocity : -args.max_velocity;
976:        }
977:
978:        // prev_velocity_l = l_velocity;
979:        // prev_velocity_r = r_velocity;
980:        //
981:        // previous_l_velocities.push_back(l_velocity);
982:        // previous_r_velocities.push_back(r_velocity);
983:        // if(previous_l_velocities.size() > velocity_history) {
984:        //     previous_l_velocities.erase(previous_l_velocities.begin());
985:        // }
986:        //
987:        // if(previous_r_velocities.size() > velocity_history) {
988:        //     previous_r_velocities.erase(previous_r_velocities.begin());
989:        // }
990:        error_history.push_back(prev_error);
991:        if(error_history.size() > max_history_length) {
992:            error_history.erase(error_history.begin());
993:        }
994:
995:
996:        std::cout << l_velocity << " " << r_velocity << " " << relative_angle << " " << error << "\n";
997:        // for(int i=0; i < previous_l_velocities.size(); i++) {
998:        //     std::cout << previous_l_velocities.at(i) << " ";
999:        // }
1000:       // std::cout << "\n";
1001:       double error_difference = *std::minmax_element(error_history.begin(), error_history.end()).second - *std::minmax_element(error_history.begin(), error_history.end()).first;
1002:
1003:       if ( args.log_data ) {
1004:           Logger logger;
1005:           log_entry entry;
1006:           entry.content = (
1007:               "[INFO] " + std::string("CHASSIS_PID_TURN")
1008:               + ", Time: " + std::to_string(pros::millis())
1009:               + ", Actual_Vol1: " + std::to_string(front_left_drive->get_actual_voltage())
1010:               + ", Actual_Vol2: " + std::to_string(front_right_drive->get_actual_voltage())
1011:               + ", Actual_Vol3: " + std::to_string(back_left_drive->get_actual_voltage())
1012:               + ", Actual_Vol4: " + std::to_string(back_right_drive->get_actual_voltage())
1013:               + ", Slew: " + std::to_string(args.motor_slew)
1014:               + ", Brake: " + std::to_string(front_left_drive->get_brake_mode())
1015:               + ", Gear: " + std::to_string(front_left_drive->get_gearset())
1016:               + ", I_max: " + std::to_string(I_max)
1017:               + ", I: " + std::to_string(integral)
```

```
1018:                 + ", kD: " + std::to_string(kD)
1019:                 + ", kI: " + std::to_string(kI)
1020:                 + ", kP: " + std::to_string(kP)
1021:                 + ", Position_Sp: " + std::to_string(0)
1022:                 + ", position_l: " + std::to_string(std::get<0>(Sensors::get_average_encoders(l_id, r_id)))
1023:                 + ", position_r: " + std::to_string(std::get<1>(Sensors::get_average_encoders(l_id, r_id)))
1024:                 + ", Heading_Sp: " + std::to_string(args.setpoint1)
1025:                 + ", Relative_Heading: " + std::to_string(relative_angle)
1026:                 + ", Absolute Angle: " + std::to_string(abs_angle)
1027:                 + ", error history: " + std::to_string(error_history.size())
1028:                 + ", history size: " + std::to_string(max_history_length)
1029:                 + ", time out time: " + std::to_string(start_time + args.timeout)
1030:                 + ", error difference: " + std::to_string(error_difference)
1031:                 + ", over slew: " + std::to_string(over_slew)
1032:                 + ", Actual_Vel1: " + std::to_string(front_left_drive->get_actual_velocity())
1033:                 + ", Actual_Vel2: " + std::to_string(front_right_drive->get_actual_velocity())
1034:                 + ", Actual_Vel3: " + std::to_string(back_left_drive->get_actual_velocity())
1035:                 + ", Actual_Vel4: " + std::to_string(back_right_drive->get_actual_velocity())
1036:             );
1037:             entry.stream = "clog";
1038:             logger.add(entry);
1039:         }
1040:
1041:         // settled is when error is almost zero and velocity is minimal
1042:         // double l_difference = *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second - *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).first;
1043:         // double r_difference = *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).second - *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).first;
1044:         // std::cout << "difference: " << *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second << " " << previous_l_velocities.size() << "\n";
1045:         if (
1046:             std::abs(error_difference) < .007
1047:             && error_history.size() == max_history_length
1048:             && pros::millis() > start_time + 500
1049:             // std::abs(l_difference) < 2
1050:             // && previous_l_velocities.size() == velocity_history
1051:             // && std::abs(r_difference) < 2
1052:             // && previous_r_velocities.size() == velocity_history
1053:             // && l_velocity < 2
1054:             // && r_velocity < 2
1055:         ) {  // velocity change has been minimal, so stop
1056:             front_left_drive->set_motor_mode(e_voltage);
1057:             front_right_drive->set_motor_mode(e_voltage);
1058:             back_left_drive->set_motor_mode(e_voltage);
1059:             back_right_drive->set_motor_mode(e_voltage);
1060:
1061:             front_left_drive->set_voltage(0);
1062:             front_right_drive->set_voltage(0);
1063:             back_left_drive->set_voltage(0);
1064:             back_right_drive->set_voltage(0);
1065:             std::cout << "ending\n";
1066:             break; // end before timeout
1067:         }
1068:
1069:         front_left_drive->move_velocity(l_velocity);
1070:         front_right_drive->move_velocity(r_velocity);
1071:         back_left_drive->move_velocity(l_velocity);
1072:         back_right_drive->move_velocity(r_velocity);
1073:
1074:
1075:         pros::delay(10);
1076:     } while ( pros::millis() < (start_time + args.timeout) );
1077:
1078:     front_left_drive->set_motor_mode(e_voltage);
1079:     front_right_drive->set_motor_mode(e_voltage);
1080:     back_left_drive->set_motor_mode(e_voltage);
1081:     back_right_drive->set_motor_mode(e_voltage);
1082:
1083:     front_left_drive->set_voltage(0);
1084:     front_right_drive->set_voltage(0);
1085:     back_left_drive->set_voltage(0);
1086:     back_right_drive->set_voltage(0);
1087:
1088:     front_left_drive->enable_driver_control();
1089:     front_right_drive->enable_driver_control();
1090:     back_left_drive->enable_driver_control();
1091:     back_right_drive->enable_driver_control();
1092:
1093:     right_encoder->forget_position(r_id);  // free up space in the encoders log
1094:     left_encoder->forget_position(l_id);
1095: }
1096:
1097:
1098:
1099: void Chassis::t_move_to_waypoint(chassis_params args, waypoint point) {
1100:     PositionTracker* tracker = PositionTracker::get_instance();
1101:
1102:     long double dx = point.x - tracker->get_position().x_pos;
1103:     long double dy = point.y - tracker->get_position().y_pos;
1104:
1105:     // convert end coordinates to polar to find the change in angle
1106:     // long double dtheta = std::fmod((-M_PI / 2) + std::atan2(dy, dx), (2 * M_PI));
1107:     long double dtheta = std::atan2(dy, dx);
1108:     if(dtheta < 0) {  // map to [0, 2pi]
1109:         dtheta += 2 * M_PI;
1110:     }
1111:
1112:     // current angle is bounded by [-pi, pi] re map it to [0, 2pi]
1113:     long double current_angle = tracker->get_heading_rad();
1114:     if(current_angle < 0) {
1115:         current_angle += 2 * M_PI;
1116:     }
1117:     current_angle = (-current_angle) + (M_PI / 2);
1118:
1119:     // calculate how much the robot needs to turn to be at the angle
1120:     long double to_turn_face_forwards = current_angle - dtheta;  // change in robot angle
1121:     long double to_turn_face_backwards = (current_angle - dtheta) - M_PI;
1122:
1123:     if(to_turn_face_forwards > M_PI) {  // find minimal angle change and direction of change [-PI/2, PI/2]
1124:         to_turn_face_forwards = (-2 * M_PI) + to_turn_face_forwards;  // give negative value to turn left to point
1125:     } else if(to_turn_face_forwards < -M_PI) {
1126:         to_turn_face_forwards = (2 * M_PI) + to_turn_face_forwards;  // give positive value to turn right to point
1127:     }
1128:
1129:     if(to_turn_face_backwards > M_PI) {  // find minimal angle change and direction of change [-PI/2, PI/2]
1130:         to_turn_face_backwards = (-2 * M_PI) + to_turn_face_backwards;  // give negative value to turn left to point
```

```
1131:     } else if(to_turn_face_backwards < -M_PI) {
1132:         to_turn_face_backwards = (2 * M_PI) + to_turn_face_backwards;  // give positive value to turn right to point
1133:     }
1134:
1135:
1136:     long double to_turn;
1137:     int direction;
1138:     if(args.explicit_direction == 1) {  // force positive direction
1139:         to_turn = to_turn_face_forwards;
1140:         direction = 1;
1141:     } else if(args.explicit_direction == -1) {  // force negative direction
1142:         to_turn = to_turn_face_backwards;
1143:         direction = -1;
1144:     } else if(std::abs(to_turn_face_forwards) < std::abs(to_turn_face_backwards)) {  // faster to go forwards
1145:         to_turn = to_turn_face_forwards;
1146:         direction = 1;
1147:     } else {  // faster to go backwards
1148:         to_turn = to_turn_face_backwards;
1149:         direction = -1;
1150:     }
1151:
1152:     to_turn = tracker->to_degrees(to_turn);
1153:
1154:
1155:     // set up turn
1156:     chassis_params turn_args;
1157:     turn_args.setpoint1 = to_turn;
1158:     turn_args.max_velocity = args.max_velocity;
1159:     turn_args.timeout = 15000; // TODO: add time estimation
1160:     args.kP = 2.8;
1161:     args.kI = 0.0005;
1162:     args.kD = 50;
1163:     args.I_max = INT32_MAX;
1164:     turn_args.motor_slew = args.motor_slew;
1165:     turn_args.log_data = args.log_data;
1166:
1167:     // perform turn
1168:     std::cout << "starting turn\n";
1169:     std::cout << to_turn << "\n";
1170:     t_turn(turn_args);
1171:     std::cout << "turn done\n";
1172:
1173:     // caclulate distance to move to point
1174:     long double distance = std::sqrt((std::pow(dx, 2) + std::pow(dy, 2)));
1175:     long double to_drive = direction * tracker->to_encoder_ticks(distance, wheel_diameter);
1176:
1177:     // set up straight drive
1178:     chassis_params drive_straight_args;
1179:     drive_straight_args.setpoint1 = to_drive;
1180:     drive_straight_args.setpoint2 = to_drive;
1181:     drive_straight_args.max_velocity = 125;
1182:     drive_straight_args.timeout = 15000;
1183:     drive_straight_args.kP = .77;
1184:     drive_straight_args.kI = 0.000002;
1185:     drive_straight_args.kD = 7;
1186:     drive_straight_args.I_max = INT32_MAX;
1187:     drive_straight_args.motor_slew = args.motor_slew;
1188:     drive_straight_args.correct_heading = args.correct_heading;
1189:     drive_straight_args.log_data = args.log_data;
1190:     //
1191:     // std::cout << "starting drive\n";
1192:     // std::cout << to_drive << "\n";
1193:     // drive_straight_args.kP = .2;
1194:     // drive_straight_args.kI = .001;
1195:     // drive_straight_args.kD = 0;
1196:     // drive_straight_args.I_max = 2000;
1197:     t_pid_straight_drive(drive_straight_args);
1198:
1199:     std::cout << "drive finished\n";
1200:     if(args.log_data) {
1201:         Logger logger;
1202:         log_entry entry;
1203:         entry.content = (
1204:             "[INFO] " + std::string("CHASSIS_ODOM")
1205:             + ", Time: " + std::to_string(pros::millis())
1206:             + ", Waypoint: " + point.get_string()
1207:             + ", ToTurnForwards: " + std::to_string(tracker->to_degrees(to_turn_face_forwards))
1208:             + ", ToTurnBackwards: " + std::to_string(tracker->to_degrees(to_turn_face_backwards))
1209:             + ", ToTurn: " + std::to_string(to_turn)
1210:             + ", ToDrive: " + std::to_string(to_drive)
1211:             + ", Direction: " + std::to_string(direction)
1212:             + ", dx: " + std::to_string(dx)
1213:             + ", dy: " + std::to_string(dy)
1214:             + ", X: " + std::to_string(tracker->get_position().x_pos)
1215:             + ", Y: " + std::to_string(tracker->get_position().y_pos)
1216:             + ", Theta: " + std::to_string(tracker->to_degrees(tracker->get_position().theta))
1217:         );
1218:         entry.stream = "clog";
1219:         logger.add(entry);
1220:     }
1221: }
1222:
1223:
1224:
1225: int Chassis::pid_straight_drive(double encoder_ticks, int relative_heading /*0*/, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, bool correct_heading /*true*/, double slew /*0.2*/, bool log_data /*false*/) {
1226:     chassis_params args;
1227:     args.setpoint1 = encoder_ticks;
1228:     args.setpoint2 = encoder_ticks;
1229:     args.max_velocity = max_velocity;
1230:     args.timeout = timeout;
1231:     args.kP = .77;
1232:     args.kI = 0.000002;
1233:     args.kD = 7;
1234:     args.I_max = INT32_MAX;
1235:     args.motor_slew = slew;
1236:     args.correct_heading = correct_heading;
1237:     args.log_data = log_data;
1238:
1239:     // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
1240:     int uid = pros::millis() * (std::abs(encoder_ticks) + 1) + max_velocity + front_left_drive->get_actual_voltage();
1241:
1242:     chassis_action command = {args, uid, e_pid_straight_drive};
1243:     while ( command_start_lock.exchange( true ) ); //aquire lock
```

```
1244:     command_queue.push(command);
1245:     command_start_lock.exchange( false ); //release lock
1246:
1247:     if(!asynch) {
1248:       wait_until_finished(uid);
1249:     }
1250:
1251:     return uid;
1252:   }
1253:
1254:   int Chassis::profiled_straight_drive(double encoder_ticks, int max_velocity  /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, bool correct_heading /*true*/, int relative_heading /*0*/, bool log_data /*false*/) {
1255:     chassis_params args;
1256:     args.setpoint1 = encoder_ticks;
1257:     args.setpoint2 = relative_heading;
1258:     args.max_velocity = max_velocity;
1259:     args.timeout = timeout;
1260:     args.kP = 2;
1261:     args.kI = 0.0005;
1262:     args.kD = 0.001;
1263:     args.I_max = INT32_MAX;
1264:     args.correct_heading = correct_heading;
1265:     args.log_data = log_data;
1266:
1267:     // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
1268:     int uid = pros::millis() * (std::abs(encoder_ticks) + 1) + max_velocity + front_left_drive->get_actual_voltage();
1269:
1270:     chassis_action command = {args, uid, e_profiled_straight_drive};
1271:     while ( command_start_lock.exchange( true ) ); //aquire lock
1272:     command_queue.push(command);
1273:     command_start_lock.exchange( false ); //release lock
1274:
1275:     if(!asynch) {
1276:       wait_until_finished(uid);
1277:     }
1278:
1279:     return uid;
1280:   }
1281:
1282:
1283:
1284:   int Chassis::uneven_drive(double l_enc_ticks, double r_enc_ticks, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*10*/, bool log_data /*false*/) {
1285:     chassis_params args;
1286:     args.setpoint1 = l_enc_ticks;
1287:     args.setpoint2 = r_enc_ticks;
1288:     args.max_velocity = max_velocity;
1289:     args.timeout = timeout;
1290:     args.kP = .77;
1291:     args.kI = 0.000002;
1292:     args.kD = 7;
1293:     args.I_max = INT32_MAX;
1294:     args.motor_slew = slew;
1295:     args.correct_heading = false;
1296:     args.log_data = log_data;
1297:
1298:     // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
1299:     int uid = pros::millis() * (std::abs(l_enc_ticks) + 1) + max_velocity + front_left_drive->get_actual_voltage();
1300:
1301:     chassis_action command = {args, uid, e_pid_straight_drive};
1302:     while ( command_start_lock.exchange( true ) ); //aquire lock
1303:     command_queue.push(command);
1304:     command_start_lock.exchange( false ); //release lock
1305:
1306:     if(!asynch) {
1307:       wait_until_finished(uid);
1308:     }
1309:
1310:     return uid;
1311:   }
1312:
1313:
1314:
1315:   int Chassis::turn_right(double degrees, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*15*/, bool log_data /*false*/) {
1316:     chassis_params args;
1317:     args.setpoint1 = degrees;
1318:     args.max_velocity = max_velocity;
1319:     args.timeout = timeout;
1320:     args.kP = 2.8;
1321:     args.kI = 0.0005;
1322:     args.kD = 50;
1323:     args.I_max = INT32_MAX;
1324:     args.motor_slew = slew;
1325:     args.log_data = log_data;
1326:
1327:     // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
1328:     int uid = pros::millis() * (std::abs(degrees) + 1) + max_velocity + front_left_drive->get_actual_voltage();
1329:
1330:     chassis_action command = {args, uid, e_turn};
1331:     while ( command_start_lock.exchange( true ) ); //aquire lock
1332:     command_queue.push(command);
1333:     command_start_lock.exchange( false ); //release lock
1334:
1335:     if(!asynch) {
1336:       wait_until_finished(uid);
1337:     }
1338:
1339:     return uid;
1340:   }
1341:
1342:
1343:
1344:   int Chassis::turn_left(double degrees, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*15*/, bool log_data /*false*/) {
1345:     chassis_params args;
1346:     args.setpoint1 = -degrees;
1347:     args.max_velocity = max_velocity;
1348:     args.timeout = timeout;
1349:     args.kP = 2.8;
1350:     args.kI = 0.0005;
1351:     args.kD = 50;
1352:     args.I_max = INT32_MAX;
1353:     args.motor_slew = slew;
1354:     args.log_data = log_data;
1355:
1356:     // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
```

```
1357:     int uid = pros::millis() * (std::abs(degrees) + 1) + max_velocity + front_left_drive->get_actual_voltage();
1358:
1359:     chassis_action command = {args, uid, e_turn};
1360:     while ( command_start_lock.exchange( true ) ); //aquire lock
1361:     command_queue.push(command);
1362:     command_start_lock.exchange( false ); //release lock
1363:
1364:     if(!asynch) {
1365:         wait_until_finished(uid);
1366:     }
1367:
1368:     return uid;
1369: }
1370:
1371:
1372: int Chassis::drive_to_point(double x, double y, int recalculations /*0*/, int explicit_direction /*0*/, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool correct_heading /*true*/, bool asynch /*false*/, double slew /*10*/, bool log_data /*true*/) {
1373:     chassis_params args;
1374:     args.setpoint1 = x;
1375:     args.setpoint2 = y;
1376:     args.max_velocity = max_velocity;
1377:     args.timeout = timeout;
1378:     args.recalculations = recalculations;
1379:     args.explicit_direction = explicit_direction;
1380:     args.motor_slew = slew;
1381:     args.correct_heading = correct_heading;
1382:     args.log_data = log_data;
1383:
1384:     // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
1385:     int uid = pros::millis() * (std::abs(x) + 1) + max_velocity + front_left_drive->get_actual_voltage();
1386:
1387:     chassis_action command = {args, uid, e_drive_to_point};
1388:     while ( command_start_lock.exchange( true ) ); //aquire lock
1389:     command_queue.push(command);
1390:     command_start_lock.exchange( false ); //release lock
1391:
1392:     if(!asynch) {
1393:         wait_until_finished(uid);
1394:     }
1395:
1396:     return uid;
1397: }
1398:
1399:
1400:
1401: int Chassis::turn_to_point(double x, double y, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*10*/, bool log_data /*true*/) {
1402:     chassis_params args;
1403:     args.setpoint1 = x;
1404:     args.setpoint2 = y;
1405:     args.max_velocity = max_velocity;
1406:     args.timeout = timeout;
1407:     args.motor_slew = slew;
1408:     args.log_data = log_data;
1409:
1410:     // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
1411:     int uid = pros::millis() * (std::abs(x) + 1) + max_velocity + front_left_drive->get_actual_voltage();
1412:
1413:     chassis_action command = {args, uid, e_turn_to_point};
1414:     while ( command_start_lock.exchange( true ) ); //aquire lock
1415:     command_queue.push(command);
1416:     command_start_lock.exchange( false ); //release lock
1417:
1418:     if(!asynch) {
1419:         wait_until_finished(uid);
1420:     }
1421:
1422:     return uid;
1423: }
1424:
1425:
1426:
1427: int Chassis::turn_to_angle(double theta, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*10*/, bool log_data /*true*/) {
1428:     PositionTracker* tracker = PositionTracker::get_instance();
1429:     chassis_params args;
1430:     args.setpoint1 = tracker->to_radians(theta);
1431:     args.max_velocity = max_velocity;
1432:     args.timeout = timeout;
1433:     args.motor_slew = slew;
1434:     args.log_data = log_data;
1435:
1436:     // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
1437:     int uid = pros::millis() * (std::abs(theta) + 1) + max_velocity + front_left_drive->get_actual_voltage();
1438:
1439:     chassis_action command = {args, uid, e_turn_to_angle};
1440:     while ( command_start_lock.exchange( true ) ); //aquire lock
1441:     command_queue.push(command);
1442:     command_start_lock.exchange( false ); //release lock
1443:
1444:     if(!asynch) {
1445:         wait_until_finished(uid);
1446:     }
1447:
1448:     return uid;
1449: }
1450:
1451:
1452: /**
1453:  * sets scaled voltage of each drive motor
1454:  */
1455: void Chassis::move( int voltage )
1456: {
1457:     front_left_drive->move(voltage);
1458:     front_right_drive->move(voltage);
1459:     back_left_drive->move(voltage);
1460:     back_right_drive->move(voltage);
1461: }
1462:
1463:
1464: /**
1465:  * sets a new brakemode for each drive motor
1466:  */
1467: void Chassis::set_brake_mode( pros::motor_brake_mode_e_t new_brake_mode )
1468: {
1469:     front_left_drive->set_brake_mode(new_brake_mode);
```

```
1470:     front_right_drive->set_brake_mode(new_brake_mode);
1471:     back_left_drive->set_brake_mode(new_brake_mode);
1472:     back_right_drive->set_brake_mode(new_brake_mode);
1473:   }
1474:
1475:
1476:
1477:
1478:   /**
1479:    * sets all chassis motors to the opposite direction that they were facing
1480:    * ie. reversed is now normal and normal is now reversed
1481:    */
1482:   void Chassis::change_direction()
1483:   {
1484:     front_left_drive->reverse_motor();
1485:     front_right_drive->reverse_motor();
1486:     back_left_drive->reverse_motor();
1487:     back_right_drive->reverse_motor();
1488:   }
1489:
1490:
1491:
1492:
1493:   /**
1494:    * sets slew to enabled for each motor
1495:    * sets the rate of the slew to the rate parameter
1496:    */
1497:   void Chassis::enable_slew( int rate /*120*/ )
1498:   {
1499:     front_left_drive->enable_slew();
1500:     front_right_drive->enable_slew();
1501:     back_left_drive->enable_slew();
1502:     back_right_drive->enable_slew();
1503:
1504:     front_left_drive->set_slew(rate);
1505:     front_right_drive->set_slew(rate);
1506:     back_left_drive->set_slew(rate);
1507:     back_right_drive->set_slew(rate);
1508:   }
1509:
1510:
1511:
1512:
1513:   /**
1514:    * sets slew to disabled for each motor
1515:    */
1516:   void Chassis::disable_slew( )
1517:   {
1518:     front_left_drive->disable_slew();
1519:     front_right_drive->disable_slew();
1520:     back_left_drive->disable_slew();
1521:     back_right_drive->disable_slew();
1522:   }
1523:
1524:
1525:   void Chassis::wait_until_finished(int uid) {
1526:     while(std::find(commands_finished.begin(), commands_finished.end(), uid) == commands_finished.end()) {
1527:       pros::delay(10);
1528:     }
1529:     while ( command_finish_lock.exchange( true ) ); //aquire lock
1530:     commands_finished.erase(std::remove(commands_finished.begin(), commands_finished.end(), uid), commands_finished.end());
1531:     command_finish_lock.exchange( false ); //release lock
1532:   }
1533:
1534:
1535:   bool Chassis::is_finished(int uid) {
1536:     if(std::find(commands_finished.begin(), commands_finished.end(), uid) == commands_finished.end()) {
1537:       while ( command_finish_lock.exchange( true ) ); //aquire lock
1538:       commands_finished.erase(std::remove(commands_finished.begin(), commands_finished.end(), uid), commands_finished.end());
1539:       command_finish_lock.exchange( false ); //release lock
1540:
1541:       return false;  // command is not finished because it is not in the list
1542:     }
1543:     return true;
1544:   }
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Jan 22 12:49:32 2021

@author: aiden
"""

tower_colors = ["blue", "blue", "red"] # top, middle, bottom
indexer_colors = ["red", "none", "none"] #top, middle, bottom

final_tower = ["red", "red", "none"]

num_balls_to_cycle = 0
tower_initial = [i for i in tower_colors if i != "none"]
indexer_initial = [i for i in tower_colors if i != "none"]
cycle = indexer_colors + tower_colors
while cycle[3:] != final_tower:
    num_balls_to_cycle += 1
    cycle.insert(0, cycle.pop())
    in_tower = cycle[3:]
    in_indexer = cycle[:3]
    print(num_balls_to_cycle, in_indexer, in_tower)
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/subsystems/intakes.hpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * Contains class for the front intakes
8:   * has methods for intaking
9:   */
10:
11: #ifndef __INTAKES_HPP__
12: #define __INTAKES_HPP__
13:
14: #include <tuple>
15: #include <queue>
16:
17: #include "main.h"
18:
19: #include "../motors/Motor.hpp"
20: #include "../sensors/Sensors.hpp"
21: #include "../sensors/BallDetector.hpp"
22:
23:
24: typedef enum e_intake_command {
25:     e_intake,
26:     e_stop_movement,
27:     e_secure,
28:     e_hold_outward,
29:     e_rocket_outwards
30: } intake_command;
31:
32:
33: /**
34:  * @see: Motors.hpp
35:  *
36:  * contains methods to allow for control of the indexer
37:  */
38: class Intakes
39: {
40:     private:
41:         static Motor *l_intake;
42:         static Motor *r_intake;
43:
44:         static int num_instances;
45:
46:         pros::Task *thread;  // the motor thread
47:         static std::queue<intake_command> command_queue;
48:         static std::atomic<bool> lock;
49:
50:         static void intake_motion_task(void*);
51:
52:     public:
53:         Intakes(Motor &left, Motor &right);
54:         ~Intakes();
55:
56:         void intake();
57:         void stop();
58:         void intake_until_secure();
59:         void hold_outward();
60:         void rocket_outwards();
61:
62:         void reset_queue();
63: };
64:
65:
66:
67:
68: #endif
```

```
1:  /**
2:   * @file: ./RobotCode/src/objects/subsystems/intakes.cpp
3:   * @author: Aiden Carney
4:   * @reviewed_on:
5:   * @reviewed_by:
6:   *
7:   * Contains implementation for the front intakes subsystem
8:   * has methods for intaking
9:   */
10:
11: #include "main.h"
12:
13:
14: #include "../serial/Logger.hpp"
15: #include "intakes.hpp"
16:
17: int Intakes::num_instances = 0;
18: std::queue<intake_command> Intakes::command_queue;
19: std::atomic<bool> Intakes::lock = ATOMIC_VAR_INIT(false);
20: Motor* Intakes::l_intake;
21: Motor* Intakes::r_intake;
22:
23: Intakes::Intakes(Motor &left, Motor &right)
24: {
25:     l_intake = &left;
26:     r_intake = &right;
27:
28:     l_intake->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
29:     r_intake->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
30:
31:     l_intake->set_motor_mode(e_voltage);
32:     r_intake->set_motor_mode(e_voltage);
33:
34:     l_intake->disable_slew();
35:     r_intake->disable_slew();
36:
37:     if(num_instances == 0 || thread == NULL) {
38:         thread = new pros::Task( intake_motion_task, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "intakes_thread");
39:     }
40:
41:     num_instances += 1;
42: }
43:
44:
45: Intakes::~Intakes() {
46:     num_instances -= 1;
47:     if(num_instances == 0) {
48:         delete thread;
49:     }
50: }
51:
52:
53:
54:
55: void Intakes::intake_motion_task(void*) {
56:     l_intake->tare_encoder();
57:     r_intake->tare_encoder();
58:     l_intake->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
59:     r_intake->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
60:
61:     int abs_position_l = 0;  // the absolute postions are calculated based on the change in encoder value
62:     int abs_position_r = 0;  // and capped to max and min values
63:     int prev_encoder_l = l_intake->get_encoder_position();
64:     int prev_encoder_r = r_intake->get_encoder_position();
65:     int integral_l = 0;
66:     int integral_r = 0;
67:     int dt = 0;
68:     int time = pros::millis();
69:
70:     while(1) {
71:         if(command_queue.empty()) {  // delay unitl there is a command in the queue
72:             pros::delay(7);
73:             continue;
74:         }
75:
76:         // take lock and get command
77:         while ( lock.exchange( true ) ); //aquire lock
78:         intake_command command = command_queue.front();
79:         command_queue.pop();
80:         lock.exchange( false ); //release lock
81:
82:         if(command != e_hold_outward) {  // reset integral if no longer holding outwards
83:             integral_l = 0;
84:             integral_r = 0;
85:         }
86:
87:         dt = pros::millis() - time;  // calculate change in time since last command
88:         time = pros::millis();
89:
90:         int d_enc_l = l_intake->get_encoder_position() - prev_encoder_l;
91:         int d_enc_r = r_intake->get_encoder_position() - prev_encoder_r;
92:         prev_encoder_l = l_intake->get_encoder_position();
93:         prev_encoder_r = r_intake->get_encoder_position();
94:         abs_position_l += d_enc_l;
95:         abs_position_r += d_enc_r;
96:
97:         // cap encoder values. This can be done because mechanical stops stop the motion of
98:         // the intakes
99:         if (abs_position_l > 0) {  // innermost value of the encoder
100:            abs_position_l = 0;
101:        }
102:
103:        if (abs_position_r > 0) {  // innermost value of the encoder
104:            abs_position_r = 0;
105:        }
106:        // std::cout << abs_position_l << " " << l_intake->get_actual_voltage() << "\n";
107:        // execute command
108:        switch(command) {
109:            case e_intake: {
110:                l_intake->set_voltage(12000);
111:                r_intake->set_voltage(12000);
112:                break;
113:            } case e_stop_movement: {
```

```
114:                l_intake->set_voltage(0);
115:                r_intake->set_voltage(0);
116:                break;
117:            } case e_secure: {
118:                l_intake->set_voltage(12000);
119:                r_intake->set_voltage(12000);
120:                if ((l_intake->get_torque() + r_intake->get_torque()) / 2 > 1) { // wait a little bit and then say ball is secure
121:                    pros::delay(300);
122:                    l_intake->set_voltage(0);
123:                    r_intake->set_voltage(0);
124:                }
125:                break;
126:            } case e_hold_outward: {  // PI controller to hold outwards
127:                // double l_error = -37 - abs_position_l;  // set first number to encoder setpoint
128:                // double r_error = -37 - abs_position_r;  // set first number to encoder setpoint
129:                //
130:                // integral_l = integral_l + (l_error * dt);
131:                // integral_r = integral_r + (r_error * dt);
132:                //
133:                // int voltage_l = (40 * l_error) + (1 * integral_l);  // set first number to kP, second number to kI
134:                // int voltage_r = (40 * r_error) + (1 * integral_r);  // set first number to kP, second number to kI
135:                // if(abs_position_l > -30) {
136:                //     l_intake->set_voltage(-5000);
137:                // } else {
138:                //     l_intake->set_voltage(-1500);  // doesn't take a lot to keep it out, so less voltage
139:                // }
140:                //
141:                // if(abs_position_r > -30) {
142:                //     r_intake->set_voltage(-5000);
143:                // } else {
144:                //     r_intake->set_voltage(-1500);  // doesn't take a lot to keep it out, so less voltage
145:                // }
146:                //
147:                l_intake->set_voltage(-3500);
148:                r_intake->set_voltage(-3500);
149:                break;
150:            } case e_rocket_outwards: {
151:                l_intake->set_voltage(-12000);
152:                r_intake->set_voltage(-12000);
153:                break;
154:            }
155:        }
156:
157:    }
158: }
159:
160: void Intakes::intake() {
161:     while ( lock.exchange( true ) ); //aquire lock
162:     command_queue.push(e_intake);
163:     lock.exchange( false ); //release lock
164: }
165:
166: void Intakes::stop() {
167:     reset_queue();
168:     while ( lock.exchange( true ) ); //aquire lock
169:     command_queue.push(e_stop_movement);
170:     lock.exchange( false ); //release lock
171: }
172:
173: void Intakes::intake_until_secure() {
174:     while ( lock.exchange( true ) ); //aquire lock
175:     command_queue.push(e_secure);
176:     lock.exchange( false ); //release lock
177: }
178:
179: void Intakes::hold_outward() {
180:     while ( lock.exchange( true ) ); //aquire lock
181:     command_queue.push(e_hold_outward);
182:     lock.exchange( false ); //release lock
183: }
184:
185: void Intakes::rocket_outwards() {
186:     while ( lock.exchange( true ) ); //aquire lock
187:     command_queue.push(e_rocket_outwards);
188:     lock.exchange( false ); //release lock
189: }
190:
191: void Intakes::reset_queue() {
192:     while ( lock.exchange( true ) ); //aquire lock
193:     std::queue<intake_command> empty_queue;
194:     std::swap( command_queue, empty_queue );  // replace command queue with an empty queue
195:     lock.exchange( false ); //release lock
196: }
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jan 17 12:47:59 2021

@author: aiden
"""


import matplotlib.pyplot as plt

def gen_profile(enc_ticks, max_acceleration, max_decceleration, max_velocity, initial_velocity):
    profile = [initial_velocity]

    i = 0
    while(i < enc_ticks):
        ticks_left = enc_ticks - i
        ticks_to_deccelerate = profile[i] / max_decceleration
        if(ticks_to_deccelerate < ticks_left):
            step = (profile[i] + max_acceleration(i))
            if(step > max_velocity):
                step = max_velocity
            profile.append(step)
        else:
            profile.append((profile[i] - max_decceleration))

        i += 1

    return profile

def accel_profile(x):
    vel = .005 * x
    print(vel)
    return vel

ticks = 1000

y = gen_profile(ticks, accel_profile, 0.8, 450, 50)
x = list(range(ticks + 1))
print(len(y))
plt.scatter(x, y)
plt.show()
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 31 14:01:59 2020

@author: aiden
"""

import kivy
from kivy.app import App
from kivy.clock import Clock
from kivy.uix.label import Label
from kivy.uix.gridlayout import GridLayout
from kivy.uix.textinput import TextInput
from kivy.uix.button import Button
from kivy.uix.widget import Widget
from kivy.properties import ObjectProperty
from kivy.uix.tabbedpanel import TabbedPanel
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.popup import Popup
from kivy.properties import StringProperty
from kivy.event import EventDispatcher
import sys


class Data:
    __instance = None

    motors = {
        0:"Front Right",
        1:"Front Left",
        2:"Back Right",
        3:"Back Left",
        4:"Main Intake",
        5:"Hoarding Intake",
        6:"Lift",
    }

    @staticmethod
    def get_instance():
        """ Static access method. """
        if Data.__instance == None:
            Data()
        return Data.__instance

    def __init__(self):
        if Data.__instance != None:
            raise RuntimeError("Constructor has already been called and exists at " + str(Data.__instance))
        else:
            Data.__instance = self
            self.motors_data = {}

    def api_interaction(self, byte1, byte2, msg):
        return None

    def retrieve_motor_data(self, *args):
        data = {}
        for i, motor in self.motors.items():
            motor_data = {
                "Actual Velocity":self.api_interaction(0xA0, 0xA0, i),
                "Actual Voltage":self.api_interaction(0xA0, 0xA1, i),
                "Current Draw":self.api_interaction(0xA0, 0xA2, i),
                "Encoder Position":self.api_interaction(0xA0, 0xA3, i),
                "Brakemode":self.api_interaction(0xA0, 0xA4, i),
                "Gearset":self.api_interaction(0xA0, 0xA5, i),
                "Port":self.api_interaction(0xA0, 0xA6, i),
                "PID Constants":self.api_interaction(0xA0, 0xA7, i),
                "Slew Rate":self.api_interaction(0xA0, 0xA8, i),
                "Power":self.api_interaction(0xA0, 0xA9, i),
                "Temperature":self.api_interaction(0xA0, 0xAA, i),
                "Torque":self.api_interaction(0xA0, 0xAB, i),
                "Direction":self.api_interaction(0xA0, 0xAC, i),
                "Efficiency":self.api_interaction(0xA0, 0xAD, i),
                "Is Stopped":self.api_interaction(0xA0, 0xAE, i),
                "Is Reversed":self.api_interaction(0xA0, 0xAF, i),
                "Is Registered":self.api_interaction(0xA1, 0xA0, i)
            }

            data.update({i:motor_data})
        self.motors_data = data


class Settings:
    __instance = None

    @staticmethod
    def get_instance():
        """ Static access method. """
        if Settings.__instance == None:
            Settings()
        return Settings.__instance

    def __init__(self):
        if Settings.__instance != None:
            raise RuntimeError("Constructor has already been called and exists at " + str(Settings.__instance))
        else:
            Settings.__instance = self
            self.ip_address = "127.0.0.1"
            self.motor_dashboard_selected = "Actual Velocity"
            self.motor_selected = 0

    def update_motor_dashboard_selected(self, new_value):
        self.motor_dashboard_selected = new_value

    def update_motor_selected(self, new_value):
        self.motor_selected = new_value
        if self.motor_selected not in Data.motors.keys():
            self.motor_selected = 0


class SettingsPopup(FloatLayout):
    def update_ip_addr(self, new_ip):
```

```
114:         print("New IP set: ", new_ip)
115:
116:
117:
118:  class MainScreen(FloatLayout):
119:
120:      motor_title_label_text = StringProperty("")
121:
122:      def __init__(self):
123:          super(MainScreen, self).__init__()
124:
125:      def open_settings(self):
126:          s = SettingsPopup()
127:          popup_window = Popup(
128:              title="Settings",
129:              content=s,
130:              size_hint=(None,None),
131:              size=(self.width / 2, self.height / 2)
132:          )
133:
134:          popup_window.open()
135:
136:      @classmethod
137:      def update_motor_info_labels(cls):
138:          # self.ids.get("label0").text = str(Data.get_instance().motors_data.get(0, {}).get(App.get_running_app().settings.motor_dashboard_selected))
139:          # self.ids.get("label1").text = str(Data.get_instance().motors_data.get(1, {}).get(App.get_running_app().settings.motor_dashboard_selected))
140:          # self.ids.get("label2").text = str(Data.get_instance().motors_data.get(2, {}).get(App.get_running_app().settings.motor_dashboard_selected))
141:          # self.ids.get("label3").text = str(Data.get_instance().motors_data.get(3, {}).get(App.get_running_app().settings.motor_dashboard_selected))
142:          # self.ids.get("label4").text = str(Data.get_instance().motors_data.get(4, {}).get(App.get_running_app().settings.motor_dashboard_selected))
143:          # self.ids.get("label5").text = str(Data.get_instance().motors_data.get(5, {}).get(App.get_running_app().settings.motor_dashboard_selected))
144:          # self.ids.get("label6").text = str(Data.get_instance().motors_data.get(6, {}).get(App.get_running_app().settings.motor_dashboard_selected))
145:
146:          cls.motor_title_label_text = str(Data.get_instance().motors.get(Settings.get_instance().motor_selected))
147:
148:
149:  class VexServer(App):
150:      settings = Settings.get_instance()
151:      l0 = StringProperty("")
152:      l1 = StringProperty("")
153:      l2 = StringProperty("")
154:      l3 = StringProperty("")
155:      l4 = StringProperty("")
156:      l5 = StringProperty("")
157:      l6 = StringProperty("")
158:      motor_data_title = StringProperty("")
159:      motor_data_body = StringProperty("")
160:
161:
162:      def update_motor_data(self, *args):
163:          self.l0 = str(Data.get_instance().motors_data.get(0, {}).get(self.settings.motor_dashboard_selected))
164:          self.l1 = str(Data.get_instance().motors_data.get(1, {}).get(self.settings.motor_dashboard_selected))
165:          self.l2 = str(Data.get_instance().motors_data.get(2, {}).get(self.settings.motor_dashboard_selected))
166:          self.l3 = str(Data.get_instance().motors_data.get(3, {}).get(self.settings.motor_dashboard_selected))
167:          self.l4 = str(Data.get_instance().motors_data.get(4, {}).get(self.settings.motor_dashboard_selected))
168:          self.l5 = str(Data.get_instance().motors_data.get(5, {}).get(self.settings.motor_dashboard_selected))
169:          self.l6 = str(Data.get_instance().motors_data.get(6, {}).get(self.settings.motor_dashboard_selected))
170:
171:          self.motor_data_title = str(Data.get_instance().motors.get(self.settings.motor_selected))
172:          body_text = ""
173:          for key, value in Data.get_instance().motors_data.get(self.settings.motor_selected, {}).items():
174:              body_text += key + ": " + str(value) + '\n'
175:          # print(self.motor_data_body)
176:          self.motor_data_body = body_text
177:
178:      def build(self):
179:          Clock.schedule_interval(self.update_motor_data, 0.1)
180:          Clock.schedule_interval(Data.get_instance().retrieve_motor_data, 0.01)
181:          return MainScreen()
182:
183:
184:  def mainloop(dt):
185:      # VexServer.set_more_info_labels(App.get_running_app().settings.motor_selected)
186:      # VexServer.set_comparison_labels(App.get_running_app().settings.motor_dashboard_selected)
187:      # print(App.get_running_app().l0)
188:      MainScreen.update_motor_info_labels()
189:
190:
191:
192:  if __name__ == "__main__":
193:      VexServer().run()
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jul 26 11:00:42 2020

@author: aiden
"""

import multiprocessing as mp
import serial
import subprocess
import threading
import time
import queue
from functools import wraps
import sys




def create_double(n):
    if n >= 0:
        sign = 0
    else:
        sign = 1

    bias = 1023
    interval = (0, 2048)
    exp_term = 0
    while not 1 <= exp_term < 2:
        mid = int(interval[0] + (interval[1] - interval[0]) / 2)
        exp_term = (abs(n) / (2**(mid - bias)))

        if exp_term < 1:  # use lower range
            interval = (interval[0], mid + 1)
            exp_guess = mid
        else:  # use upper range
            mid = int(interval[0] + (interval[1] - interval[0]) / 2)
            interval = (mid - 1, interval[1])
            exp_guess = mid

    significand = 0
    total = exp_term - 1
    for i in range(1, 53):
        if total - (2**(-i)) >= 0:
            total = total - (2**(-i))
            significand  |= 2**(52 - i)

    byte_list = []
    # first 6 bytes are from significand
    for i in range(6):
        byte = significand & 0xff
        byte_list.append(byte)
        significand = significand >> 8

    # 7th byte is part significand (4 bits) and part exponent (4 bits)
    byte = significand
    byte |= ((exp_guess & 0x0f) << 4)
    exp_guess = exp_guess >> 4
    byte_list.append(byte)

    # 8th byte is part exponent (7 bits) and the sign bit (1 bit)
    byte = exp_guess & 0x7f
    byte |= sign << 7
    byte_list.append(byte)

    return byte_list



class Client:
    def __init__(self, uid):
        self.uid = uid
        self.send_queue = queue.Queue()
        self.send_queue_lock = threading.Lock()

        self.recv_queue = queue.Queue()
        self.recv_queue_lock = threading.Lock()


    def _send_message(self, id1, id2, msg=""):
        msg = id1 + id2 + msg
        with self.send_queue_lock:
            self.send_queue.put(msg)


    def _receive_message(self, max_wait=5, sent_message=""):
        if max_wait is None:
            max_wait = sys.maxsize - 1

        start = time.time()
        while 1:  # set max waiting time to 5 sec
            if not self.recv_queue.empty():
                with self.recv_queue_lock:
                    received = self.recv_queue.get()
                break

            end = time.time()

            if (end - start) > max_wait:
                error_msg = "No response returned from host in allotted time"
                if sent_message:
                    error_msg += " with sent message: " + sent_message
                raise TimeoutError(error_msg)

        return received


    def get_command(self, id1, id2, msg=""):
        self._send_message(id1, id2, msg)
        return self._receive_message(5, msg)

    def post_command(self, id1, id2, msg=""):
```

```
114:         pass
115:
116:     def debug(self, debug_message):
117:         self._send_message('\xAB', '\xA0', debug_message)
118:         return self._receive_message(5, debug_message)
119:
120:
121: class ServerConnection:
122:     def __init__(self, debug=False, read_chunk_size=1024):
123:         self.connection = None
124:         self.debug = debug
125:         self.read_chunk_size = read_chunk_size
126:
127:         self.__write_thread = threading.Thread(target=self.write_thread)
128:         self.run_writing_thread = False
129:
130:         self.__read_thread = threading.Thread(target=self.read_server_stdout)
131:         self.run_reading_thread = False
132:
133:         self.__write_thread.daemon = True
134:         self.__read_thread.daemon = True
135:
136:         self.clients = []
137:         self.client_lock = threading.Lock()
138:
139:         self.connection_lock = threading.Lock()
140:
141:         self.__write_thread.start()
142:         self.__read_thread.start()
143:
144:
145:
146:     def serial_exception_handler(func):
147:         @wraps(func)
148:         def inner_function(self, *args, **kwargs):
149:             while 1:
150:                 try:
151:                     return func(self, *args, **kwargs)
152:                 except serial.SerialException as e:
153:                     if self.debug:
154:                         print(e)
155:                 except serial.serialutil.SerialException as e:
156:                     if self.debug:
157:                         print(e)
158:                 except OSError as e:
159:                     if self.debug:
160:                         print(e)
161:                 except AttributeError:
162:                     if self.debug:
163:                         print("Connection is not established; attemting to establish one")
164:
165:                 self.run_writing_thread = False
166:                 self.run_reading_thread = False
167:                 time.sleep(.5)
168:                 self.connection = None
169:
170:                 with self.connection_lock:  # use a lock in case multiple threads are trying to establish a connection
171:                     while not self.mount_vex_brain():
172:                         time.sleep(.1)
173:                         if self.debug:
174:                             print("retrying connection", flush=True)
175:
176:                 self.run_writing_thread = True
177:                 self.run_reading_thread = True
178:
179:
180:         return inner_function
181:
182:
183:     def mount_vex_brain(self):
184:         command = "./usb.sh"
185:         process = subprocess.Popen(command, stdout=subprocess.PIPE)
186:
187:         ttys = []
188:         for i in process.stdout.readlines():
189:             i = i.decode("utf-8")
190:             if "VEX" in i:
191:                 ttys.append(i.split(" ")[0])
192:
193:         if not ttys:
194:             if self.debug:
195:                 print("No mount points for the vex brain were found", flush=True)
196:             return 0
197:
198:         mnts = sorted(ttys, reverse=True)
199:         tty = ""
200:         for i in mnts:
201:             if "ACM" in i:
202:                 tty = i
203:                 break
204:
205:         try:
206:             self.connection = serial.Serial(
207:                 tty,
208:                 baudrate=115200,
209:                 bytesize=serial.EIGHTBITS,
210:                 parity=serial.PARITY_NONE,
211:                 stopbits=serial.STOPBITS_ONE
212:             )
213:
214:         except serial.SerialException:
215:             if self.debug:
216:                 print("Failed to open Vex Brain on ", tty)
217:             return 0
218:
219:         if self.debug:
220:             print("connection established")
221:
222:         return 1
223:
224:
225:     @serial_exception_handler
226:     def read_bytes(self):
```

```
227:         return self.connection.read(self.connection.in_waiting)
228:
229:     @serial_exception_handler
230:     def write_bytes(self, send_array):
231:         self.connection.write(send_array)
232:         return 1
233:
234:
235:     def read_server_stdout(self):
236:         read_check = 0
237:         while 1:
238:             if self.run_reading_thread:
239:                 bytes_read = iter(self.read_bytes())
240:                 terminal_output = ""
241:                 for byte in bytes_read:
242:                     if read_check == 0 and byte == 0xAA:
243:                         read_check = 1
244:                     elif read_check == 1 and byte == 0x55:
245:                         read_check = 2
246:                     elif read_check == 2 and byte == 0x1E:
247:                         read_check = 3
248:                     elif read_check == 3:
249:                         num_bytes_following = byte
250:                         uid_msb = next(bytes_read)
251:                         uid_lsb = next(bytes_read)
252:                         uid = (uid_msb << 8) | uid_lsb
253:                         msg = ""
254:                         for _ in range(num_bytes_following - 2):
255:                             try:
256:                                 char = chr(next(bytes_read))
257:                             except UnicodeDecodeError:
258:                                 if self.debug:
259:                                     print("failed to decode character")
260:                                 char = ""
261:
262:                             msg += char
263:                         if self.debug:
264:                             print("message received: ", msg, "at", time.time())
265:
266:                         checksum = next(bytes_read)
267:                         if checksum == 0xC6:
268:                             # find server with that id and add message to its queue
269:                             for client in self.clients:
270:                                 if client.uid == uid:
271:                                     with client.recv_queue_lock:
272:                                         client.recv_queue.put(msg)
273:
274:                         elif checksum != 0xC6 and self.debug:
275:                             print("checksum failed - received: ", checksum)
276:
277:                         read_check = 0
278:
279:                     else:  # if response from server is not part of message send to stdout
280:                         read_check = 0
281:                         try:
282:                             char = chr(byte)
283:                         except UnicodeDecodeError:
284:                             print(byte)
285:                             char = ""
286:                         terminal_output += char
287:                         # print(char, end="")
288:                 with open("log.txt", "a") as f:
289:                     f.write(terminal_output)
290:                     terminal_output = ""
291:
292:                 time.sleep(.1)
293:
294:
295:     def write_thread(self):
296:         while 1:
297:             if self.run_writing_thread:
298:                 send_array = bytearray()
299:                 for client in self.clients:
300:                     with client.send_queue_lock:
301:                         if not client.send_queue.empty():
302:                             to_write = client.send_queue.get()
303:                         else:
304:                             continue
305:
306:
307:                     send_array.append(0xAA)
308:                     send_array.append(0x55)
309:                     send_array.append(0x1E)
310:                     send_array.append(len(to_write) + 2)  # add two for the uid bytes
311:
312:                     send_array.append((client.uid >> 8) & 0xFF)
313:                     send_array.append(client.uid & 0xFF)
314:
315:                     for i in to_write:
316:                         send_array.append(ord(i))
317:
318:                     send_array.append(0xC6)
319:
320:                     if self.debug:
321:                         print("Message added to be sent: ", to_write, "at", time.time())
322:
323:                 if send_array:
324:                     self.write_bytes(send_array)
325:                     if self.debug:
326:                         print("Message array sent at", time.time())
327:
328:                     # write garbage on the stream to help clear any blocking functions on server
329:                 # send_array = bytearray()
330:                 # send_array.append(0xFF)
331:                 # send_array.append(0xFF)
332:                 # send_array.append(0xFF)
333:                 # send_array.append(0xFF)
334:                 # send_array.append(0xFF)
335:
336:                 # self.write_bytes(send_array)
337:
338:                 # time.sleep(.01)
339:
```

```
340:
341:     def add_clients(self, *args):
342:         with self.client_lock:
343:             for client in args:
344:                 self.clients.append(client)
345:
346:
347:     def start_server(self):
348:         self.run_writing_thread = True
349:         self.run_reading_thread = True
350:
351:     def stop_server(self):
352:         self.run_writing_thread = False
353:         self.run_reading_thread = False
354:
355:
356: def handle_requests_async(connection, *args, **kwargs):
357:     clients = []
358:     for i in range(55000, 55000 + len(args)):
359:         client = Client(i)
360:         clients.append(client)
361:
362:     connection.add_clients(*clients)
363:
364:     for request, client in zip(args, clients):
365:         client._send_message(request[0], request[1], request[2])
366:
367:     responses = [None for i in range(len(args))]
368:     i = 0
369:     start = time.time()
370:     dt = 0
371:     while None in responses and dt < kwargs.get("max_wait", 5):
372:         try:
373:             response = clients[i]._receive_message(max_wait=.001)
374:             responses[i] = response
375:         except TimeoutError:
376:             pass
377:         i += 1
378:
379:         if i > len(clients) - 1:
380:             i = 0
381:         dt = time.time() - start
382:
383:     return responses
384:
385:
386:
387: if __name__ == "__main__":
388:     c = ServerConnection(debug=True)
389:     x = c.mount_vex_brain()
390:     c.start_server()
391:     client = Client(55000)
392:     c.add_client(client)
393:     while 1:
394:         print("starting debug msg at", time.time())
395:         client.get_command('\xA0', '\xA0', "0")
396:         # start = time.time()
397:         # print("start time:", start)
398:         # for i in range(20):
399:         #     client._send_message('\xAB', '\xA0', "test")
400:
401:         # print("time to send 20 messages: ", time.time() - start)
402:         print("ended debug msg at", time.time())
403:         time.sleep(20)
404:
405:
406:
407:
408:
409:
410:
411:
412:
413:
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Aug  6 14:30:19 2020

@author: aiden
"""


from kivy.app import App
from kivy.properties import StringProperty
from kivy.uix.floatlayout import FloatLayout
from kivy.clock import Clock
import random
import time as t

class MainScreen(FloatLayout):
    pass

class TestApp(App):
    time = ""

    def update(self, *args):
        self.time = str(t.asctime()) # + 'time'?

    def build(self):
        Clock.schedule_interval(self.update, 1)
        return MainScreen()


if __name__ == "__main__":
    TestApp().run()

```

```
#!/bin/bash

for sysdevpath in $(find /sys/bus/usb/devices/usb*/ -name dev); do

        syspath="${sysdevpath%/dev}"
        devname="$(udevadm info -q name -p $syspath)"
        [[ "$devname" == "bus/"* ]] && continue
        eval "$(udevadm info -q property --export -p $syspath)"
        [[ -z "$ID_SERIAL" ]] && continue
        echo "/dev/$devname - $ID_SERIAL"

done
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 27 16:48:08 2020

@author: aiden
"""
import flask

import serial_client


# values = """0xA0 0xA0
# 0xA0 0xA1
# 0xA0 0xA2
# 0xA0 0xA3
# 0xA0 0xA4
# 0xA0 0xA5
# 0xA0 0xA6
# 0xA0 0xA7
# 0xA0 0xA8
# 0xA0 0xA9
# 0xA0 0xAA
# 0xA0 0xAB
# 0xA0 0xAC
# 0xA0 0xAD
# 0xA0 0xAE
# 0xA0 0xAF
# 0xA1 0xA0"""

# for line in values.split("\n"):
#     byte_s = line.split(" ")
#     msb = byte_s[0].strip()
#     lsb = byte_s[1].strip()

#     return_id = (int(msb, 0) << 8) | int(lsb, 0);
#     print(return_id)
motors = {
    0:"Front Right",
    1:"Front Left",
    2:"Back Right",
    3:"Back Left",
    4:"Main Intake",
    5:"Hoarding Intake",
    6:"Lift",
}

app = flask.Flask(__name__)


def get_motor_data(connection, motor_num):
    data = serial_client.handle_requests_async(connection,
        ('\xA0', '\xA0', motor_num),
        ('\xA0', '\xA1', motor_num),
        ('\xA0', '\xA2', motor_num),
        ('\xA0', '\xA3', motor_num),
        ('\xA0', '\xA4', motor_num),
        ('\xA0', '\xA5', motor_num),
        ('\xA0', '\xA6', motor_num),
        ('\xA0', '\xA7', motor_num),
        ('\xA0', '\xA8', motor_num),
        ('\xA0', '\xA9', motor_num),
        ('\xA0', '\xAA', motor_num),
        ('\xA0', '\xAB', motor_num),
        ('\xA0', '\xAC', motor_num),
        ('\xA0', '\xAD', motor_num),
        ('\xA0', '\xAE', motor_num),
        ('\xA0', '\xAF', motor_num),
        ('\xA1', '\xA0', motor_num)
    )
    motor_data = {
        "Actual Velocity":data[0],
        "Actual Voltage":data[1],
        "Current Draw":data[2],
        "Encoder Position":data[3],
        "Brakemode":data[4],
        "Gearset":data[5],
        "Port":data[6],
        "PID Constants":data[7],
        "Slew Rate":data[8],
        "Power":data[9],
        "Temperature":data[10],
        "Torque":data[11],
        "Direction":data[12],
        "Efficiency":data[13],
        "Is Stopped":data[14],
        "Is Reversed":data[15],
        "Is Registered":data[16]
    }
        # motor_data = {
        #     "Actual Velocity":client.get_command('\xA0', '\xA0', motor_num),
        #     "Actual Voltage":client.get_command('\xA0', '\xA1', motor_num),
        #     "Current Draw":client.get_command('\xA0', '\xA2', motor_num),
        #     "Encoder Position":client.get_command('\xA0', '\xA3', motor_num),
        #     "Brakemode":client.get_command('\xA0', '\xA4', motor_num),
        #     "Gearset":client.get_command('\xA0', '\xA5', motor_num),
        #     "Port":client.get_command('\xA0', '\xA6', motor_num),
        #     "PID Constants":client.get_command('\xA0', '\xA7', motor_num),
        #     "Slew Rate":client.get_command('\xA0', '\xA8', motor_num),
        #     "Power":client.get_command('\xA0', '\xA9', motor_num),
        #     "Temperature":client.get_command('\xA0', '\xAA', motor_num),
        #     "Torque":client.get_command('\xA0', '\xAB', motor_num),
        #     "Direction":client.get_command('\xA0', '\xAC', motor_num),
        #     "Efficiency":client.get_command('\xA0', '\xAD', motor_num),
        #     "Is Stopped":client.get_command('\xA0', '\xAE', motor_num),
        #     "Is Reversed":client.get_command('\xA0', '\xAF', motor_num),
        #     "Is Registered":client.get_command('\xA1', '\xA0', motor_num)
        # }
    # except TimeoutError as e:
        # print(e)
        # motor_data = {
        #     "Actual Velocity":None,
        #     "Actual Voltage":None,
```

```
114:         #     "Current Draw":None,
115:         #     "Encoder Position":None,
116:         #     "Brakemode":None,
117:         #     "Gearset":None,
118:         #     "Port":None,
119:         #     "PID Constants":None,
120:         #     "Slew Rate":None,
121:         #     "Power":None,
122:         #     "Temperature":None,
123:         #     "Torque":None,
124:         #     "Direction":None,
125:         #     "Efficiency":None,
126:         #     "Is Stopped":None,
127:         #     "Is Reversed":None,
128:         #     "Is Registered":None
129:         # }
130:
131:     return motor_data
132:
133:
134: class InvalidUsage(Exception):
135:     status_code = 400
136:
137:     def __init__(self, message, status_code=None, payload=None):
138:         Exception.__init__(self)
139:         self.message = message
140:         if status_code:
141:             self.status_code = status_code
142:         self.payload = payload
143:
144:     def to_dict(self):
145:         rv = dict(self.payload or ())
146:         rv['message'] = self.message
147:         return rv
148:
149:
150: @app.errorhandler(InvalidUsage)
151: def handle_invalid_usage(error):
152:     response = flask.jsonify(error.to_dict())
153:     response.status_code = error.status_code
154:     return response
155:
156:
157: @app.route("/api/motor_data/<motor_number>", methods=["GET"])
158: def api_get_motor_data(motor_number):
159:     if int(motor_number) in motors.keys():
160:         data = get_motor_data(server_conn, motor_number)
161:         return flask.jsonify(data)
162:
163:     else:
164:         raise InvalidUsage("Motor Number supplied was not valid", status_code=406)
165:
166: # @app.route("/api/debug", methods=["GET"])
167: # def api_debug():
168: #     motor_client.debug("test message")
169:
170:
171: server_conn = serial_client.ServerConnection(debug=True)
172: x = server_conn.mount_vex_brain()
173: server_conn.start_server()
174:
175:
176:
177: app.run(host='0.0.0.0')
178:
179:
180:
181:
182:
183:
184:
185:
186:
```