David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278

Lab 2 - Report

Assignment 1:
Answer the following questions in your lab report:
1. With all arguments set at default (nothing specified in the command line), which launchfiles and from which packages will be included? Which will be skipped?

Package	Launchfiles	Included/Skipped
gazebo_ros	gzserver.launch.py	Included
gazebo_ros	gzclient.launch.py	Included
turtlebot3_gazebo	robot_state_publisher.launch.py	Included
turtlebot3_gazebo	spawn_turtlebot3.launch.py	Included
prob_rob_labs	flaky_door_opener_launch.py	Skipped
vision	video_processor_launch.py	Skipped
prob_rob_labs	image_mean_feature_x_launch.py	Skipped

2. Which Launch file and from which package will be included if run_door_opener:=true is appended to the ros2 launch command above?

The flaky_door_opener_launch.py launch file from the prob_rob_labs packages will now be included.

3. At which coordinates in the XY plane will the robot be spawned if no arguments are specified?

x_pose = LaunchConfiguration('x_pose', default='-1.5')
y_pose = LaunchConfiguration('y_pose', default='0.0')

The robot will be spawned at (x, y) = (-1.5, 0) if no arguments are specified.

4. What would the Launch command look like if you were to spawn the robot at coordinates(-5, 1)?

ros2 launch prob_rob_labs turtlebot3_and_door_launch.py x_pose:=-5 y_pose:=1

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278

Assignment 2: Examine the plugin code and answer the following in your lab report:

1. In which line of which SDF file is the joint called “hinge” that the plugin controls specified and what is its type?

In line 132 in the file models/models/hinged_glass_door/model.sdf, the joint called “hinge” is specified and it is of type revolute.

```
<joint name="hinge" type="revolute">
```

2. What is the name and mass of the link that hangs off the joint that the plugin is Controlling?

The name of the link that hangs off the joint is “door” and its mass is 41.3256.

3. The model name associated with the door is hinged_glass_door. What is the name and the type of the topic to which you must publish to open or close the door? Hint: look at the plugin source code. Even if you are not an expert in C++, you should be able to figure out where the code subscribes to the topic. Then use ros2 topic list and ros2 topic info to examine the topic and make sure you got it right. You can also use the ros2 node list and ros2 node info to examine the nodes.

The name of the topic is /hinged_glass_door/torque and the type is Float64 data

4. Use the ros2 topic pub command to open the door. Hint: Type ros2 topic pub –help first to understand the command syntax and also read about the ros2-topic utility at https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Topics/Understanding-ROS2-Topics.html

ros2 topic pub --once /hinged_glass_door/torque std_msgs/msg/Float64 "{data: 1.0}"

5. What is the minimum torque you had to use for the door to open
The minimum torque we had to use to open the door is 1.0.

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278

Assignment 3: Add the code to the boilerplate node that you have just created to perform the following task:
1. Open the door.
2. Move the robot through the door.
3. Stop the robot once it went through the door.
4. Close the door behind.

Code for assignment 3:

```
import rclpy
from rclpy.node import Node

from std_msgs.msg import Float64
from geometry_msgs.msg import Twist

heartbeat_period = 0.1

class OpenMoveCloseStop(Node):

    def __init__(self):
        super().__init__('open_move_close_stop')
        self.publisher_door = self.create_publisher(Float64,
'/hinged_glass_door/torque', 10)
        self.publisher_cmd_vel = self.create_publisher(Twist,
'/cmd_vel', 10)
        self.log = self.get_logger()
        self.timer = self.create_timer(heartbeat_period,
self.heartbeat)
        self.start_time = None

    def heartbeat(self):
        if self.start_time is None:
            self.start_time =
self.get_clock().now().seconds_nanoseconds()[0]
            return

        t = self.get_clock().now().seconds_nanoseconds()[0] -
self.start_time

        torque = Float64()
        twist = Twist()
```

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278

```
        if t < 10.0:
            torque.data = 2.0
            self.publisher_door.publish(torque)
            self.log.info('Opening door')

        elif t < 20.0:
            twist.linear.x = 0.5
            self.publisher_cmd_vel.publish(twist)
            self.log.info('Moving through door')

        elif t < 25.0:
            twist.linear.x = 0.0
            self.publisher_cmd_vel.publish(twist)
            self.log.info('Stopping robot')

        elif t < 35.0:
            torque.data = -2.0
            self.publisher_door.publish(torque)
            self.log.info('Closing door')

        elif t < 40.0:
            torque.data = 0.0
            self.publisher_door.publish(torque)
            self.log.info('Finished!')

    def spin(self):
        rclpy.spin(self)

def main():
    rclpy.init()
    open_move_close_stop = OpenMoveCloseStop()
    open_move_close_stop.spin()
    open_move_close_stop.destroy_node()
    rclpy.shutdown()

if __name__ == '__main__':
    main()
```

Video Screencast:
https://youtu.be/CxI4hQJ01Wg

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278

Assignment 4: Modify the code from Assignment 3 to parametrize the forward speed commanded to the robot. Submit the code.

Code of our Node open_move_stop_close:

```
import rclpy
from rclpy.node import Node

from std_msgs.msg import Float64
from geometry_msgs.msg import Twist

heartbeat_period = 0.1

class OpenMoveCloseStop(Node):

    def __init__(self):
        super().__init__('open_move_close_stop')
        self.publisher_door = self.create_publisher(Float64,
'/hinged_glass_door/torque', 10)
        self.publisher_cmd_vel = self.create_publisher(Twist,
'/cmd_vel', 10)
        self.log = self.get_logger()
        self.timer = self.create_timer(heartbeat_period,
self.heartbeat)
        self.start_time = None

        self.declare_parameter('forward_speed', 0.5)
        self.declare_parameter('open_torque', 2.0)
        self.declare_parameter('close_torque', -2.0)

        self.declare_parameter('open_time', 10.0)
        self.declare_parameter('move_time', 10.0)
        self.declare_parameter('stop_time', 5.0)
        self.declare_parameter('close_time', 10.0)

        # Precompute time thresholds
        self.open_time   = self.get_parameter('open_time').value
        self.move_time   = self.get_parameter('move_time').value
        self.stop_time   = self.get_parameter('stop_time').value
        self.close_time  = self.get_parameter('close_time').value
```

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278

```
        self.state_times = [
            self.open_time,
            self.move_time,
            self.stop_time,
            self.close_time,
        ]
        self.state_cumulative_times = [sum(self.state_times[:i+1])
for i in range(len(self.state_times))]

    def heartbeat(self):
        if self.start_time is None:
            self.start_time =
self.get_clock().now().seconds_nanoseconds()[0]
            return

        t = self.get_clock().now().seconds_nanoseconds()[0] -
self.start_time

        forward_speed =
self.get_parameter('forward_speed').get_parameter_value().double_valu
e
        open_torque   =
self.get_parameter('open_torque').get_parameter_value().double_value
        close_torque  =
self.get_parameter('close_torque').get_parameter_value().double_value

        torque = Float64()
        twist = Twist()

        if t < self.state_cumulative_times[0]:
            torque.data = open_torque
            self.publisher_door.publish(torque)
            self.log.info('Opening door')

        elif t < self.state_cumulative_times[1]:
            twist.linear.x = forward_speed
            self.publisher_cmd_vel.publish(twist)
            self.log.info(f'Moving through door at speed
{forward_speed}')

        elif t < self.state_cumulative_times[2]:
            twist.linear.x = 0.0
            self.publisher_cmd_vel.publish(twist)
```

```
            self.log.info('Stopping robot')

        elif t < self.state_cumulative_times[3]:
            torque.data = close_torque
            self.publisher_door.publish(torque)
            self.log.info('Closing door')

        else:
            torque.data = 0.0
            self.publisher_door.publish(torque)
            self.log.info('Finished!')


    def spin(self):
        rclpy.spin(self)


def main():
    rclpy.init()
    open_move_close_stop = OpenMoveCloseStop()
    open_move_close_stop.spin()
    open_move_close_stop.destroy_node()
    rclpy.shutdown()


if __name__ == '__main__':
    main()
```

Code of our node launch file:

```
import os

from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument
from launch.substitutions import LaunchConfiguration
from launch_ros.actions import Node


def generate_launch_description():
    return LaunchDescription([
        DeclareLaunchArgument('use_sim_time', default_value='true',
                              description='set to true for
simulation'),
```

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278

```
        DeclareLaunchArgument('forward_speed', default_value='0.5',
                              description='Forward speed of the
robot'),

        DeclareLaunchArgument('open_torque', default_value='2.0',
                              description='Torque to apply when
opening the door'),

        DeclareLaunchArgument('close_torque', default_value='-2.0',
                              description='Torque to apply when
closing the door'),

        DeclareLaunchArgument('open_time', default_value='10.0',
                              description='Time to keep applying open
torque'),

        DeclareLaunchArgument('move_time', default_value='10.0',
                              description='Time to move forward'),

        DeclareLaunchArgument('stop_time', default_value='5.0',
                              description='Time to keep robot
stopped'),

        DeclareLaunchArgument('close_time', default_value='10.0',
                              description='Time to keep applying
close torque'),

        Node(
            package='prob_rob_labs',
            executable='open_move_close_stop',
            name='open_move_close_stop',
            parameters=[{
                'use_sim_time': LaunchConfiguration('use_sim_time'),
                'forward_speed':
LaunchConfiguration('forward_speed'),
                'open_torque': LaunchConfiguration('open_torque'),
                'close_torque': LaunchConfiguration('close_torque'),
                'open_time': LaunchConfiguration('open_time'),
                'move_time': LaunchConfiguration('move_time'),
                'stop_time': LaunchConfiguration('stop_time'),
                'close_time': LaunchConfiguration('close_time')
            }]
        )
```

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278

```
    ])
```

Assignment 5: Run your program multiple times and vary the speed parameter, while examining the odometry topic, until you determine the maximum velocity that the robot can achieve. Report the result. Submit the output of the echo command.

Command:

```
ros2 launch prob_rob_labs open_move_close_stop_launch.py
forward_speed:=3.7 move_time:=5.0
```

The maximum velocity that the robot can achieve is 3.65.

Output of the echo command:

2.3365090068055845e-05

0.00010650886802802904

0.0001900794273652028

0.00026509059967766326

0.00033008885968907907

0.0003755385707756341 4

0.0003924692932457424

0.00040403662871044555

0.0003745326205049593

0.00031726328531656906

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
0.00024723691418811097

0.00015251945069979933

5.647946677771265e-05

-3.981289975968293e-05

-0.00011668282116257036

-0.00017616931380173012

-0.00017961093428786034

-0.00019282586880887444

-0.00015783933175961498

-9.979817516578418e-05

-1.0952738341700186e-05

7.243758496058518e-05

0.00017151926552671056

0.0352820255671078

0.06958699184381979

0.10362706231442435

0.13759189088977006

0.17162501454454618

0.2056448805050069

0.2395203364455841

0.2735143906640206

0.3072658284577996

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
0.3411830682879941

0.3750800962689731

0.40901229141410805

0.44293901759449383

0.4768824216398057

0.5108179955428636

0.5447688895456569

0.5786060524038709

0.612917970437467

0.6473769965508127

0.6811056045169125

0.7151885140290833

0.7493998825500413

0.7835125010351024

0.8178143215126598

0.8518012060156591

0.8858724425210582

0.9199722210011734

0.9540632370711325

0.9882047277123037

1.021773818905936

1.0555957572081416

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
1.0894789032996455

1.1233483866023317

1.157234505741715

1.1911059861831237

1.2249812109632696

1.2588668266178105

1.292732772573887

1.3248183441962995

1.3602202561873107

1.395326602816097

1.4299316721331752

1.4630472617613741

1.4966986601520147

1.5311786607546138

1.5655775702956656

1.5996197009257829

1.6335907747698837

1.6678007067935137

1.7019887887686083

1.736099246290847

1.769936089190592

1.8039244430324584

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
1.8376741959846188

1.8714870853849757

1.9053717191720538

1.9392385188745562

1.9731270344486447

2.007001766830873

2.0408709940529834

2.073389258579873

2.107486675322376

2.143049547472862

2.177822138415606

2.2112640423817056

2.244770525484129

2.279228990388984

2.3134018800899026

2.3475674733727514

2.381596800216497

2.4155998548469717

2.449699417751826

2.4837888786826574

2.5178480446526224

2.5519427153390315

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
2.585889408624032

2.6194743568684546

2.653354182546857

2.6872386701078237

2.7211181873005095

2.754995754636793

2.787859266109246

2.8212049354590607

2.8570753880720003

2.8915478044817844

2.9255353594168874

2.9589847354553394

2.992972663145083

3.027563779702496

3.061522079971165

3.0954435809843033

3.1295382423767646

3.163927163322256

3.1978324817456656

3.2317279664103298

3.2657631400758036

3.299769157509255

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
3.3234894594684223

3.3463669547713026

3.3695282146316283

3.3925200978572447

3.4155616378459825

3.4386358842627263

3.4608505569840022

3.484875288947696

3.5084181876815426

3.533608464888118

3.556337523611609

3.579487539964707

3.6033766382987427

3.629942681277923

3.656584928085906

3.658492443739464

3.628759590942848

3.6184976079882523

3.6126517811435095

3.609716111275886

3.6055208905077425

3.6043943802630465

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
3.603769691236562

3.6036960583239925

3.6036150861991505

3.6035168573273966

3.60340302692692

3.6033068815317844

3.6037877265222

3.6044610636707892

3.606351741548009

3.6075172557754884

3.639501928679788

3.645409606602682

3.636146279165947

3.6130437832179054

3.607419544580596

3.6071408291509046

3.6050480995984238

3.604228648382822

3.6043851008800143

3.6047787565675056

3.6051775754328697

3.604301146874579

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
3.6034936111552907

3.6033769679043104

3.6033514047208084

3.6033827695605503

3.6033421623904993

3.6034535740982356

3.60354321096819

3.602981555256327

3.6031750905834246

3.604290705947653

3.6045725735252 02

3.604281012999457

3.604527499611584

3.6051748859484416

3.605310984852829

3.6049967473829456

3.60476400065391

3.604365554935229

3.6047677288816953

3.60503574243499

3.6041990343279013

3.603712231406036

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
3.6036223996090384

3.603541464142429

3.6034303506868013

3.6033956253852604

3.6034249234475877

3.603381791156185

3.603188323953123

3.6037898615130617

3.6060991897065353

3.607919080532124

3.6072878531841672

3.607188580388419

3.6077924292526964

3.609526050864591

3.612868634137834

3.618310117557431

3.6246030194437124

3.637690422168263

3.6477575170393224

3.6221498983962594

3.5987472119816246

3.5756721184972764

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
3.552455715806539

3.529241252810583

3.506007118056413

3.482760238088609

3.4603096809923777

3.4371475348584712

3.4137655056166913

3.3904841277102675

3.3673608582153918

3.3438675449038273

3.3207128502792966

3.2968908992564456

3.262708587599347

3.2286548666624615

3.1948896731171135

3.160946898988066

3.1269585251875984

3.092967719560549

3.0590049003853466

3.025020296989482

2.9910273615105694

2.9570522282426577

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
2.923064337616251

2.889077381008638

2.855070928530574

2.8210725805575505

2.7870723096740644

2.753045310291036

2.719053346812636

2.6850324098629517

2.651039437257549

2.6170392466135226

2.583042721118401

2.5490393397011193

2.515041120945375

2.4810406870037007

2.447035600326207

2.413046526505041

2.379046294275876

2.3450479868553566

2.311054263596683

2.2770686144948047

2.243063917255716

2.2090681506236747

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
2.175070085265297

2.1410637245334416

2.107040999798577

2.073036859653631

2.0390389441916477

2.0050120780833027

1.9710262478249356

1.9370397461034803

1.9030299707460265

1.8690453091331223

1.8350575923494041

1.8010497951790296

1.767065407657787

1.7330712304291347

1.6990816704589544

1.6650808907779484

1.6310596511561253

1.5970579303575665

1.5630489774703564

1.5290346391597953

1.495056597097744

1.4610568956658747

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
1.427051100459038

1.393031837834853

1.3590523291921104

1.3250505325340667

1.2910614066313089

1.2570621199484355

1.2230568819857042

1.1890731882801704

1.1550790713593275

1.1210593214411213

1.0870764869058218

1.0530696742321979

1.0190566515845898

0.9850513133400557

0.9510356907785158

0.9170401764463032

0.883036926479981

0.8490395611600505

0.815056016283971

0.7810501097462859

0.7470627979130066

0.7130671705644211

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
0.6790808340759324

0.6450812794449412

0.6110769987319798

0.5770600144161738

0.5430527376887768

0.5090483269123596

0.47504829687081696

0.4410364532586187

0.40704882102793216

0.37304695221203504

0.3390521923415157

0.30504109826612735

0.2710407717610703

0.23704140594501646

0.20305421063289267

0.16905374081620728

0.13506103608365544

0.1010701455111206

0.06706422998139554

0.03307310738634563

-0.0009291456048145509

-4.359776871236437e-05

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
-2.249283467176142e-05

7.603866707555034e-05

3.283684426565013e-05

-4.270716460707301e-05

-3.816212303422615e-05

-3.4114359141954124e-05

7.94327645169663e-06

2.9597633082307953e-05

4.6882515543341784e-05

8.508221423603589e-05

0.0001380114528869521

0.0001942052886238803

0.0002562967691080823

0.0003100785680948232

0.0003735940020180222

0.00039677517534819135

0.00040957508449109643

0.0004121975147974517

0.00040131854237754485

0.0003726580051162775 7

0.00033413149015353857

0.0002618124087408185

David Reyes - dr3362
Prahlad Vivek - pv2391
Xincheng Ma - xm2278
0.00016683902419175413

7.043891082601234e-05

-2.5928119599907336e-05

-0.00012211341779570388

-0.00017319778867121838

-0.00018389100970956702

-0.00017155456551269394

-0.00014110226152245494

-9.189704906922452e-05

-1.5333773999606056e-05

6.909465653142622e-05

0.00015247712769053

0.00024825132746875985

0.00030183974159177544

0.00035877657376043006

0.00039018769665757 1

0.00039765923418791415

0.00037180161410682204

0.0003307563630247891

0.00027589283439918 6
