ros_diff_drive

Generated by Doxygen 1.8.17

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

Module Documentation

5.1 Pre initial values of PID regulators

Values of PID parameters before the update from the dynamic reconfigure module. Advice is to keep current values.

Variables

- float **config.KP_ROT** = 0.0
- float **config.TI_ROT** = 0.000001
- float **config.TD_ROT** = 0.0
- float **config.INT_LIMIT_ROT** = 0.0
- float **config.P_ANG_DST** = 0.0
- float **config.P_ANG_THT** = 0.0
- float **config.KP_FWD** = 0.0
- float **config.TI_FWD** = 0.000001
- float **config.TD_FWD** = 0.0
- float **config.P_FWD_DST** = 0.0
- float **config.P_FWD_CUR** = 0.0
- float **config.INT_LIMIT_FWD** = 0.0

5.1.1 Detailed Description

Values of PID parameters before the update from the dynamic reconfigure module. Advice is to keep current values.

5.2 Values updated during the callback of the odometry

Variables

- float move_to_point.xInitial = 0.0

  Initial x coordinate from the start of moving forward.
- float move_to_point.yInitial = 0.0

  Initial y coordinate from the start of moving forward.

5.2.1 Detailed Description

5.3 Values updated during the callback of the odometry

These values are updated during position_callback.

Variables

- move_to_point.cur_pos = Point()

  Contains X and Y coordinates of the current position.
- **move_to_point.x**
- **move_to_point.y**
- float move_to_point.theta = 0.0

  Contanes current angle of the robot.

5.3.1 Detailed Description

These values are updated during position_callback.

5.4 Variables used for debugging purposes

If debugging only rotation/forward, they receive the value from the dynamic reconfigure.

Variables

- float move_to_point.GOAL_THETA = 0.0

  Goal angle.
- float move_to_point.GOAL_DIST = 0.0

  Goal distance.

5.4.1 Detailed Description

If debugging only rotation/forward, they receive the value from the dynamic reconfigure.

Chapter 6

Namespace Documentation

6.1 config Namespace Reference

Variables

- float **controller_freq** = 50.0
- float **rot_speed_limit** = 2.0
- float **fwd_speed_limit** = 0.7
- float **angle_err_tolerance** = 0.04
- float **dist_err_tolerance** = 0.005
- float **KP_ROT** = 0.0
- float **TI_ROT** = 0.000001
- float **TD_ROT** = 0.0
- float **INT_LIMIT_ROT** = 0.0
- float **P_ANG_DST** = 0.0
- float **P_ANG_THT** = 0.0
- float **KP_FWD** = 0.0
- float **TI_FWD** = 0.000001
- float **TD_FWD** = 0.0
- float **P_FWD_DST** = 0.0
- float **P_FWD_CUR** = 0.0
- float **INT_LIMIT_FWD** = 0.0

6.1.1 Detailed Description

```
Static configuration of the move_to_point module
```

6.2 debug Namespace Reference

Variables

- **pub_dbg_angle_err** = rospy.Publisher("debug/angle_err", Float32, queue_size = 5)
- **pub_dbg_theta** = rospy.Publisher("debug/theta", Float32, queue_size=5)
- **pub_dbg_theta_filtr** = rospy.Publisher("debug/theta_filtr", Float32, queue_size=5)
- **pub_dbg_ang_to_goal** = rospy.Publisher("debug/angle_to_goal", Float32, queue_size=5)
- **pub_dbg_ang_to_goal_filtr** = rospy.Publisher("debug/angle_to_goal_filtr", Float32, queue_size=5)
- **pub_dbg_rot** = rospy.Publisher("debug/rot_vel", Float32, queue_size=5)
- **pub_dbg_distance** = rospy.Publisher("/debug/distance", Float32, queue_size = 5)
- **pub_dbg_distance_filtr** = rospy.Publisher("/debug/dist_filtr", Float32, queue_size = 5)
- **pub_dbg_dist_to_goal** = rospy.Publisher("/debug/dist_to_goal", Float32, queue_size = 5)
- **pub_dbg_dist_to_goal_filtr** = rospy.Publisher("/debug/dist_to_goal_filtr", Float32, queue_size = 5)
- **pub_dbg_fwd** = rospy.Publisher("/debug/dist_velocity", Float32, queue_size = 5)
- **pub_dbg_fwd_rot** = rospy.Publisher("/debug/fwd_rot", Float32, queue_size = 5)
- **pub_dbg_fwd_rot_vel** = rospy.Publisher("/debug/fwd_rot_vel", Float32, queue_size = 5)

6.2.1 Detailed Description

```
Contains debug variables and publisher/subscriber definitions
```

6.3 fsm Namespace Reference

Classes

- class FsmRobot
- class FsmState
- class FsmStates

6.3.1 Detailed Description

```
@package fsm
Finite State Machine library
```

6.4 move_to_point Namespace Reference

Functions

- def angle_error_calc (target_angle, current_angle)

  Calculate rotation direction
  Positive rotation direction - clockwise
  Negative rotation direction - counter clockwise.
- def dyn_reconf_callback (config, level)

  Dynamic reconfigure callback function.
- def position_callback (msg)

  Odometry subscriber callback function.
- def goal_position_callback (msg)

  Callback function for processing goal values.
- def idle ()

  Idle function of the robot state machine.
- def rotate ()

  Rotation function of the robot state machine.
- def forward ()

  State machine functionality for moving forward.

Variables

- float T = 1.0 / controller_freq

  Controller's period in seconds.
- float xInitial = 0.0

  Initial x coordinate from the start of moving forward.
- float yInitial = 0.0

  Initial y coordinate from the start of moving forward.
- cur_pos = Point()

  Contains X and Y coordinates of the current position.
- **x**
- **y**
- float theta = 0.0

  Contanes current angle of the robot.
- goal = Point(0, 0, 0)

  Goal input coordinates Updated in goal_position_callback.
- float GOAL_THETA = 0.0

  Goal angle.
- float GOAL_DIST = 0.0

  Goal distance.
- active_goal = Point(0, 0, 0)

  Goal coordinates that are under processing.
- float angle_to_goal_filt = 0.0

  Filtered value of the goal angle.
- float theta_filt = 0.0

  Filtered value of the current angle.
- bool POSITIVE = True

  Constant for indicating positive angle.
- bool NEGATIVE = False

  Constant for indicating negative angle.
- bool theta_sign = POSITIVE

  Variable indicating signess of the current iteration angle: POSITIVE or NEGATIVE.
- bool theta_sign_prev = POSITIVE

  Variable indicating signess of the angle from the previous iteration: POSITIVE or NEGATIVE.
- float goal_distance = 0.0

  Calculated goal distance.
- float goal_distance_filt = 0.0

  Filtered value of the goal distance.
- float dist_filt = 0.0

  Filtered value of the desired distance.
- bool move_fwd_started = False

  Indicates if moving forward started or not.
- float angle_to_goal_fwd = 0.0

  Used for correcting angle error which accumulates while moving forward.
- sub_goal_position = rospy.Subscriber("/target_position/position", Pose2D, goal_position_callback)

  Goal destination subscriber.
- sub_odom = rospy.Subscriber("/m2xr_diff_drive_controller/odom", Odometry, position_callback)

  Odometry (current position) subscriber.
- pub_cmd_vel = rospy.Publisher("/m2xr_diff_drive_controller/cmd_vel", Twist, queue_size=1)

  cmd_vel publisher Used to publish desired velocity to the next layer of the control.
- r = rospy.Rate(controller_freq)

Initialization of the speed_controller node.

- StateIdle = FsmState(FsmStates.Idle, idle)

  FSM idle state definition.
- StateRotation = FsmState(FsmStates.Rotating, rotate)

  FSM rotation state definition.
- StateForward = FsmState(FsmStates.Forward, forward)

  FSM moving forward state definition.
- list StatesList = [StateIdle, StateRotation, StateForward]

  List of permitted states.
- robot_fsm = FsmRobot("M2XR", StatesList, StatesList[0])

  FSM Initialization.
- srv__dyn_reconf = Server(DynRecPIDConfig, dyn_reconf_callback)

  Dynamic reconfigure server initialization.
- rot_pid = Regulator(KP_ROT, TI_ROT, TD_ROT, T, rot_speed_limit, INT_LIMIT_ROT)

  Normal rotation PID regulator initialization.
- fwd_pid = Regulator(KP_FWD, TI_FWD, TD_FWD, T, fwd_speed_limit, INT_LIMIT_FWD)

  Moving forward PID regulator initialization.
- fwd_pid_rot = Regulator(KP_ROT, TI_ROT, TD_ROT, T, rot_speed_limit, INT_LIMIT_ROT)

  Rotation while moving forward PID regulator initialization.

6.4.1 Detailed Description

```
Implements differential drive robot control
```

6.4.2 Function Documentation

6.4.2.1 angle_error_calc()

```
def move_to_point.angle_error_calc (
              target_angle,
              current_angle )
```

Calculate rotation direction
Positive rotation direction - clockwise
Negative rotation direction - counter clockwise.

Parameters

target_angle	Angle to be reached [degrees]
current_angle	Current angle of the robot [degrees]

Returns

Error including direction [degrees]

6.4.2.2 dyn_reconf_callback()

```
def move_to_point.dyn_reconf_callback (
              config,
              level )
```

Dynamic reconfigure callback function.

Parameters

config	Contains all dynamic parameters
level	Not used

Returns

config

6.4.2.3 forward()

```
def move_to_point.forward ( )
```

State machine functionality for moving forward.

This state controls robot when moving forward. It filters current distance and desired distance values, calculates an error, and generates desired linear speed calculated in PID routine. Desired rotation speed is then published to the velocity publisher pub_cmd_vel.

6.4.2.4 goal_position_callback()

```
def move_to_point.goal_position_callback (
              msg )
```

Callback function for processing goal values.

Parameters

msg	Message to be processed - contains desired goal position

6.4.2.5 idle()

```
def move_to_point.idle ( )
```

Idle function of the robot state machine.

In this state robot is waiting for the new command to arrive.

6.4.2.6 position_callback()

```
def move_to_point.position_callback (
              msg )
```

Odometry subscriber callback function.

Parameters

msg	The message base on Odometry message type

6.4.2.7 rotate()

```
def move_to_point.rotate ( )
```

Rotation function of the robot state machine.

This state controls robot rotation. It filters current angle and desired angle values, calculates an error in degrees and generates desired rotation speed calculated in PID routine. Desired rotation speed is then published to the velocity publisher pub_cmd_vel.

6.5 regulator Namespace Reference

Classes

- class Regulator

6.5.1 Detailed Description

```
@package regulator
Implements PID regulation algorithms
```

Chapter 7

Class Documentation

7.1 fsm.FsmRobot Class Reference

Public Member Functions

- def __init__ (self, name, states_list, state)
- def switch_state (self, new_state)
- def **validate_state** (self, state)
- def **default** (self)
- def **execute** (self)

Public Attributes

- **name**
- **states_list**
- **current_state**
- **previous_state**

7.1.1 Detailed Description

```
Finite State Machine class
Provides data structure for FSM as well as main
methods for normal functioning and events logging
```

7.1.2 Constructor & Destructor Documentation

7.1.2.1 __init__()

```
def fsm.FsmRobot.__init__ (
              self,
              name,
              states_list,
              state )
```

```
Default constructor for FSM
name: Name of the FSM
states_list: List of all available states
state: initial state
```

7.1.3 Member Function Documentation

7.1.3.1 switch_state()

```
def fsm.FsmRobot.switch_state (
              self,
              new_state )
```

```
Method used for switching between states of the FSM
```

The documentation for this class was generated from the following file:

- /home/djordje/catkin_ws/src/ros_diff_drive/scripts/fsm.py

7.2 fsm.FsmState Class Reference

Public Member Functions

- def **__init__** (self, state, method)

Public Attributes

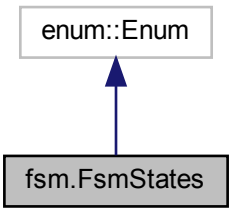
- **state**
- **method**

The documentation for this class was generated from the following file:

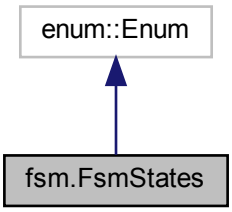
- /home/djordje/catkin_ws/src/ros_diff_drive/scripts/fsm.py

7.3 fsm.FsmStates Class Reference

Inheritance diagram for fsm.FsmStates:



Collaboration diagram for fsm.FsmStates:



Static Public Attributes

- int **Default** = 0
- int **Idle** = 1
- int **Rotating** = 2
- int **Forward** = 3

7.3.1 Detailed Description

```
FSM states enumeration
```

The documentation for this class was generated from the following file:

- /home/djordje/catkin_ws/src/ros_diff_drive/scripts/fsm.py

7.4 regulator.Regulator Class Reference

Public Member Functions

- def **__init__** (self, KP, TI, TD, T, u_limit, ui_limit)
- def **update_params** (self, KP, TI, TD, ui_limit)
- def **pid_positional** (self, error)
- def **pid_incremental** (self, error)

Public Attributes

- **KP**
- **KI**
- **KD**
- **u_limit**
- **ui_limit**
- **err_prev**
- **err_p_prev**
- **u**
- **T**
- **KDT**
- **KIT**
- **Ui**
- **error_p_prev**

The documentation for this class was generated from the following file:

- /home/djordje/catkin_ws/src/ros_diff_drive/scripts/regulator.py

Index