Solutions for Unit 5 Path Planning

Index:

Solution Exercise 5.2

Exercise 5.2

Launch File: my_move_base_launch_1.launch

In [ ]:
```
<?xml version="1.0"?>
<launch>

  <!-- Run the map server -->
  <arg name="map_file" default="$(find husky_navigation)/maps/my_map.yaml
  <node name="map_server" pkg="map_server" type="map_server" args="$(arg

  <!--- Run AMCL -->
  <include file="$(find husky_navigation)/launch/amcl.launch" />

  <!--- Run Move Base -->
  <include file="$(find my_move_base_launcher)/launch/my_move_base_launch

</launch>
```

END Launch File: my_move_base_launch_1.launch

Launch File: my_move_base_launch_2.launch

In [ ]:
```
<?xml version="1.0"?>
<launch>

  <arg name="no_static_map" default="false"/>

  <arg name="base_global_planner" default="navfn/NavfnROS"/>
  <arg name="base_local_planner" default="teb_local_planner/TebLocalPlann
  <!-- <arg name="base_local_planner" default="base_local_planner/Traject

  <node pkg="move_base" type="move_base" respawn="false" name="move_base"

    <param name="base_global_planner" value="$(arg base_global_planner)"/
    <param name="base_local_planner" value="$(arg base_local_planner)"/>
    <rosparam file="$(find my_move_base_launcher)/params/my_move_base_par

    <!-- observation sources located in costmap_common.yaml -->
    <rosparam file="$(find husky_navigation)/config/costmap_common.yaml"
    <rosparam file="$(find husky_navigation)/config/costmap_common.yaml"

    <!-- local costmap, needs size -->
    <rosparam file="$(find husky_navigation)/config/costmap_local.yaml" c
    <param name="local_costmap/width" value="10.0"/>
    <param name="local_costmap/height" value="10.0"/>

    <!-- static global costmap, static map provides size -->
    <rosparam file="$(find my_move_base_launcher)/params/my_global_costma

    <!-- global costmap with laser, for odom_navigation_demo -->
    <rosparam file="$(find husky_navigation)/config/costmap_global_laser.
    <param name="global_costmap/width" value="100.0" if="$(arg no_static_
    <param name="global_costmap/height" value="100.0" if="$(arg no_static
  </node>

</launch>
```

END Launch File: my_move_base_launch_2.launch

Solution Exercise 5.3

Exercise 5.3

Launch File: my_move_base_params.yaml

In [ ]:
```
controller_frequency: 5.0
recovery_behaviour_enabled: true

NavfnROS:
  allow_unknown: true # Specifies whether or not to allow navfn to create
  default_tolerance: 0.1 # A tolerance on the goal point for the planner.

TrajectoryPlannerROS:
  # Robot Configuration Parameters
  acc_lim_x: 2.5
  acc_lim_theta:  3.2

  max_vel_x: 1.0
  min_vel_x: 0.0

  max_vel_theta: 1.0
  min_vel_theta: -1.0
  min_in_place_vel_theta: 0.2

  holonomic_robot: false
  escape_vel: -0.1

  # Goal Tolerance Parameters
  yaw_goal_tolerance: 0.1
  xy_goal_tolerance: 0.2
  latch_xy_goal_tolerance: false

  # Forward Simulation Parameters
  sim_time: 2.0
  sim_granularity: 0.02
  angular_sim_granularity: 0.02
  vx_samples: 6
  vtheta_samples: 20
  controller_frequency: 20.0

  # Trajectory scoring parameters
  meter_scoring: true # Whether the gdist_scale and pdist_scale parameter
  occdist_scale:  0.1 #The weighting for how much the controller should a
  pdist_scale: 0.75  #     The weighting for how much the controller shou
  gdist_scale: 1.0 #     The weighting for how much the controller should

  heading_lookahead: 0.325  #How far to look ahead in meters when scoring
  heading_scoring: false  #Whether to score based on the robot's heading
  heading_scoring_timestep: 0.8   #How far to look ahead in time in secon
  dwa: true #Whether to use the Dynamic Window Approach (DWA)_ or whether
  simple_attractor: false
  publish_cost_grid_pc: true

  # Oscillation Prevention Parameters
  oscillation_reset_dist: 0.25 #How far the robot must travel in meters b
```

```
  escape_reset_dist: 0.1
  escape_reset_theta: 0.1

DWAPlannerROS:
  # Robot configuration parameters
  acc_lim_x: 2.5
  acc_lim_y: 0
  acc_lim_th: 3.2

  max_vel_x: 0.5
  min_vel_x: 0.0
  max_vel_y: 0
  min_vel_y: 0

  max_trans_vel: 0.5
  min_trans_vel: 0.1
  max_rot_vel: 1.0
  min_rot_vel: 0.2

  # Goal Tolerance Parameters
  yaw_goal_tolerance: 0.1
  xy_goal_tolerance: 1.0
  latch_xy_goal_tolerance: false
```

END Launch File: my_move_base_params.yaml

Solution Exercise 5.4

Exercise 5.4

Launch File: my_move_base_params.yaml

In [ ]:
```
controller_frequency: 5.0
recovery_behaviour_enabled: true

NavfnROS:
  allow_unknown: true # Specifies whether or not to allow navfn to create
  default_tolerance: 0.1 # A tolerance on the goal point for the planner.

TrajectoryPlannerROS:
  # Robot Configuration Parameters
  acc_lim_x: 2.5
  acc_lim_theta:  3.2
```

```
  max_vel_x: 1.0
  min_vel_x: 0.0

  max_vel_theta: 1.0
  min_vel_theta: -1.0
  min_in_place_vel_theta: 0.2

  holonomic_robot: false
  escape_vel: -0.1

  # Goal Tolerance Parameters
  yaw_goal_tolerance: 0.1
  xy_goal_tolerance: 0.2
  latch_xy_goal_tolerance: false

  # Forward Simulation Parameters
  sim_time: 2.0
  sim_granularity: 0.02
  angular_sim_granularity: 0.02
  vx_samples: 6
  vtheta_samples: 20
  controller_frequency: 20.0

  # Trajectory scoring parameters
  meter_scoring: true # Whether the gdist_scale and pdist_scale parameter
  occdist_scale:  0.1 #The weighting for how much the controller should a
  pdist_scale: 0.75  #     The weighting for how much the controller shou
  gdist_scale: 1.0 #     The weighting for how much the controller should

  heading_lookahead: 0.325  #How far to look ahead in meters when scoring
  heading_scoring: false  #Whether to score based on the robot's heading
  heading_scoring_timestep: 0.8   #How far to look ahead in time in secon
  dwa: true #Whether to use the Dynamic Window Approach (DWA)_ or whether
  simple_attractor: false
  publish_cost_grid_pc: true

  # Oscillation Prevention Parameters
  oscillation_reset_dist: 0.25 #How far the robot must travel in meters b
  escape_reset_dist: 0.1
  escape_reset_theta: 0.1

DWAPlannerROS:
  # Robot configuration parameters
  acc_lim_x: 2.5
  acc_lim_y: 0
  acc_lim_th: 3.2

  max_vel_x: 0.5
  min_vel_x: 0.0
  max_vel_y: 0
```

```
  min_vel_y: 0

  max_trans_vel: 0.5
  min_trans_vel: 0.1
  max_rot_vel: 1.0
  min_rot_vel: 0.2

  # Goal Tolerance Parameters
  yaw_goal_tolerance: 0.1
  xy_goal_tolerance: 1.0
  latch_xy_goal_tolerance: false

  sim_time: 4.0
```

END Launch File: my_move_base_params.yaml

Solution Exercise 5.7

Exercise 5.7

Launch File: my_move_base_launch_1.launch

In [ ]:
```
<?xml version="1.0"?>
<launch>

  <!-- Run the map server -->
  <arg name="map_file" default="$(find husky_navigation)/maps/my_map.yaml
  <node name="map_server" pkg="map_server" type="map_server" args="$(arg

  <!--- Run AMCL -->
  <include file="$(find husky_navigation)/launch/amcl.launch" />

  <!--- Run Move Base -->
  <include file="$(find my_move_base_launcher)/launch/my_move_base_launch

</launch>
```

END Launch File: my_move_base_launch_1.launch

Launch File: my_move_base_launch_2.launch

In [ ]:
```
<?xml version="1.0"?>
<launch>

  <arg name="no_static_map" default="false"/>

  <arg name="base_global_planner" default="navfn/NavfnROS"/>
  <arg name="base_local_planner" default="dwa_local_planner/DWAPlannerROS
  <!-- <arg name="base_local_planner" default="base_local_planner/Traject

  <node pkg="move_base" type="move_base" respawn="false" name="move_base"

    <param name="base_global_planner" value="$(arg base_global_planner)"/
    <param name="base_local_planner" value="$(arg base_local_planner)"/>
    <rosparam file="$(find my_move_base_launcher)/params/my_move_base_par

    <!-- observation sources located in costmap_common.yaml -->
    <rosparam file="$(find husky_navigation)/config/costmap_common.yaml"
    <rosparam file="$(find husky_navigation)/config/costmap_common.yaml"

    <!-- local costmap, needs size -->
    <rosparam file="$(find my_move_base_launcher)/params/my_local_costmap
    <param name="local_costmap/width" value="10.0"/> # Change to 5 for th
    <param name="local_costmap/height" value="10.0"/> # Change to 5 for t

    <!-- static global costmap, static map provides size -->
    <rosparam file="$(find my_move_base_launcher)/params/my_global_costma

    <!-- global costmap with laser, for odom_navigation_demo -->
    <rosparam file="$(find husky_navigation)/config/costmap_global_laser.
    <param name="global_costmap/width" value="100.0" if="$(arg no_static_
    <param name="global_costmap/height" value="100.0" if="$(arg no_static
  </node>

</launch>
```

END Launch File: my_move_base_launch_2.launch

Launch File: my_local_costmap_params.yaml

In [ ]:
```
global_frame: odom
rolling_window: true

plugins:
  - {name: obstacles_laser,          type: "costmap_2d::ObstacleLayer"}
  - {name: inflation,                type: "costmap_2d::InflationLayer"}
```

END Launch File: my_local_costmap_params.yaml

Solution Exercise 5.8

Exercise 5.8

Launch File: my_local_costmap_params.yaml

In [ ]:
```
global_frame: odom
rolling_window: true
update_frequency: 1.0

plugins:
  - {name: obstacles_laser,          type: "costmap_2d::ObstacleLayer"}
  - {name: inflation,                type: "costmap_2d::InflationLayer"}
```

END Launch File: my_local_costmap_params.yaml

Solution Exercise 5.9

Exercise 5.9

Launch File: my_move_base_launch_1.launch

In [ ]:
```
<?xml version="1.0"?>
<launch>

  <!-- Run the map server -->
  <arg name="map_file" default="$(find husky_navigation)/maps/my_map.yaml
  <node name="map_server" pkg="map_server" type="map_server" args="$(arg

  <!--- Run AMCL -->
  <include file="$(find husky_navigation)/launch/amcl.launch" />

  <!--- Run Move Base -->
  <include file="$(find my_move_base_launcher)/launch/my_move_base_launch

</launch>
```

END Launch File: my_move_base_launch_1.launch

Launch File: my_move_base_launch_2.launch

In [ ]:
```
<?xml version="1.0"?>
<launch>

  <arg name="no_static_map" default="false"/>

  <arg name="base_global_planner" default="navfn/NavfnROS"/>
  <arg name="base_local_planner" default="dwa_local_planner/DWAPlannerROS
  <!-- <arg name="base_local_planner" default="base_local_planner/Traject

  <node pkg="move_base" type="move_base" respawn="false" name="move_base"

    <param name="base_global_planner" value="$(arg base_global_planner)"/
    <param name="base_local_planner" value="$(arg base_local_planner)"/>
    <rosparam file="$(find my_move_base_launcher)/params/my_move_base_par

    <!-- observation sources located in costmap_common.yaml -->
    <rosparam file="$(find my_move_base_launcher)/params/my_common_costma
    <rosparam file="$(find my_move_base_launcher)/params/my_common_costma

    <!-- local costmap, needs size -->
    <rosparam file="$(find my_move_base_launcher)/params/my_local_costmap
    <param name="local_costmap/width" value="10.0"/> # Change to 5 for th
    <param name="local_costmap/height" value="10.0"/> # Change to 5 for t

    <!-- static global costmap, static map provides size -->
    <rosparam file="$(find my_move_base_launcher)/params/my_global_costma

    <!-- global costmap with laser, for odom_navigation_demo -->
    <rosparam file="$(find husky_navigation)/config/costmap_global_laser.
    <param name="global_costmap/width" value="100.0" if="$(arg no_static_
    <param name="global_costmap/height" value="100.0" if="$(arg no_static
  </node>

</launch>
```

END Launch File: my_move_base_launch_2.launch

Launch File: my_common_costmap_params.yaml

In [ ]:
```
footprint: [[-0.5, -0.33], [-0.5, 0.33], [0.5, 0.33], [0.5, -0.33]]
footprint_padding: 0.01

robot_base_frame: base_link
update_frequency: 4.0
publish_frequency: 3.0
transform_tolerance: 0.5

resolution: 0.05

obstacle_range: 1.0
raytrace_range: 6.0

#layer definitions
static:
    map_topic: /map
    subscribe_to_updates: true

obstacles_laser:
    observation_sources: laser
    laser: {data_type: LaserScan, clearing: true, marking: true, topic: s

inflation:
    inflation_radius: 1.0
```

END Launch File: my_common_costmap_params.yaml

Solution Exercise 5.10

Exercise 5.10

Launch File: my_common_costmap_params.yaml

In [ ]:
```
footprint: [[-0.5, -0.33], [-0.5, 0.33], [0.5, 0.33], [0.5, -0.33]]
footprint_padding: 0.01

robot_base_frame: base_link
update_frequency: 4.0
publish_frequency: 3.0
transform_tolerance: 0.5

resolution: 0.05

obstacle_range: 5.5
raytrace_range: 6.0

#layer definitions
static:
    map_topic: /map
    subscribe_to_updates: true

obstacles_laser:
    observation_sources: laser
    laser: {data_type: LaserScan, clearing: true, marking: true, topic: s

inflation:
    inflation_radius: 0.5
```

END Launch File: my_common_costmap_params.yaml