Section 7

Problem 1: Test these launch files with your robot sim and paste the contents of your my_nav.launch and my_nav_slow.launch files into your submission.

my_nav.launch

In [ ]:
```
<launch>
  <arg name="v_max" default="0.2"/>
  <arg name="om_max" default="0.4"/>

  <include file="$(find asl_turtlebot)/launch/turtlebot3_nav_sim.launch" />

  <node pkg="section5" type="marker_pub.py" name="marker_pub" output="screen" />

  <node pkg="asl_turtlebot" type="navigator.py" name="turtlebot_navigator" output="screen">
    <param name="v_max" value="$(arg v_max)"/>
    <param name="om_max" value="$(arg om_max)"/>
  </node>

  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find section5)/rviz/my_nav.rviz"/>

</launch>
```

my_nav_slow.launch

In [ ]:
```
<launch>
  <include file="$(find section5)/launch/my_nav.launch">
    <arg name="v_max" value="0.1"/>
    <arg name="om_max" value="0.2"/>
  </include>
</launch>
```

Problem 2: Test this on your robot sim and paste the contents of your navigator.cfg and navigator.py files into your submission.

navigator.cfg

In [ ]:
```
#!/usr/bin/env python
PACKAGE = "asl_turtlebot"

from dynamic_reconfigure.parameter_generator_catkin import *

gen = ParameterGenerator()

gen.add("k1",       double_t,    0, "Pose Controller k1", 0.8,
0., 2.0)
gen.add("k2",       double_t,    0, "Pose Controller k2", 0.4,
0., 2.0)
gen.add("k3",       double_t,    0, "Pose Controller k3", 0.4,
0., 2.0)

gen.add("spline_alpha",       double_t,    0, "trajectory smooth
ing spline_alpha", 0.12,  0., 0.2)
gen.add("traj_dt",       double_t,    0, "trajectory smoothing t
raj_dt", 0.05,  0., 0.15)

exit(gen.generate(PACKAGE, "navigator", "Navigator"))
```

navigator.py changes:

In [1]:
```
class Navigator:
    """
    This node handles point to point turtlebot motion, avoiding
obstacles.
    It is the sole node that should publish to cmd_vel
    """
    def __init__(self):

        #   ...

        # Robot limits
        self.v_max = rospy.get_param("~v_max")    # 0.2    # max
imum velocity
        self.om_max = rospy.get_param("~om_max")   # 0.4   # max
imum angular velocity

        #   ...


    def dyn_cfg_callback(self, config, level):
        rospy.loginfo("Reconfigure Request: k1:{k1}, k2:{k2}, k
3:{k3}, spline_alpha:{spline_alpha}, traj_dt:{traj_dt}".format(*
*config))
        self.pose_controller.k1 = config["k1"]
        self.pose_controller.k2 = config["k2"]
        self.pose_controller.k3 = config["k3"]

        # the new configuration:
        self.spline_alpha = config["spline_alpha"]
        self.traj_dt = config["traj_dt"]

        return config
```