Problem 1: What message type did you choose for these messages? Include your updated

code in your submission.

Message type chosen: Float64

Code:

import numpy as np

import math

from utils import wrapToPi

import rospy

from std\_msgs.msg import Float64

# command zero velocities once we are this close to the goal

RHO\_THRES = 0.05

ALPHA\_THRES = 0.1

DELTA\_THRES = 0.1

class PoseController:

    """ Pose stabilization controller """

    def \_\_init\_\_(self, k1, k2, k3, V\_max=0.5, om\_max=1):

        self.k1 = k1

        self.k2 = k2

        self.k3 = k3

        self.V\_max = V\_max

        self.om\_max = om\_max

        self.init\_publisher()

    ###############################Section 6 Publisher##############################

    def init\_publisher(self):

        # rospy.init\_node('controller',anonymous=True)

        self.pub\_alpha = rospy.Publisher('controller/alpha',Float64,queue\_size=10)

        self.pub\_delta = rospy.Publisher('controller/delta',Float64,queue\_size=10)

        self.pub\_rho   = rospy.Publisher('controller/rho',  Float64,queue\_size=10)

        self.alpha\_msg = Float64()

        self.delta\_msg = Float64()

        self.rho\_msg   = Float64()

        # rate = rospy.Rate(1)

    #############################End Section 6 Publisher############################

    def load\_goal(self, x\_g, y\_g, th\_g):

        """ Loads in a new goal position """

        self.x\_g = x\_g

        self.y\_g = y\_g

        self.th\_g = th\_g

    def compute\_control(self, x, y, th, t):

        """

        Inputs:

            x,y,th: Current state

            t: Current time (you shouldn't need to use this)

        Outputs:

            V, om: Control actions

        Hints: You'll need to use the wrapToPi function. The np.sinc function

        may also be useful, look up its documentation

        """

        ########## Code starts here ##########

        rho   = np.sqrt((x-self.x\_g)\*\*2+(y-self.y\_g)\*\*2)

        alpha = wrapToPi(np.arctan2((self.y\_g-y),(self.x\_g-x)) - th)

        delta = wrapToPi(alpha+th-self.th\_g)

        V     = self.k1\*rho\*np.cos(alpha)

        om    = self.k2\*alpha + self.k1\*np.sinc(alpha/np.pi)\*np.cos(alpha)\*(alpha+self.k3\*delta)

        self.alpha\_msg.data , self.delta\_msg.data , self.rho\_msg.data = [alpha,delta,rho]

        self.pub\_alpha.publish(self.alpha\_msg)

        self.pub\_delta.publish(self.delta\_msg)

        self.pub\_rho.publish(self.rho\_msg)

        ########## Code ends here ##########

        # apply control limits

        V = np.clip(V, -self.V\_max, self.V\_max)

        om = np.clip(om, -self.om\_max, self.om\_max)

        return V, om

Problem 2: What command did you use to record the requested topics to a particular \_le

name?

rosbag record -o Section6

Problem 3: Take a screenshot of the resulting plot in rqt plot and include it in your submission.

