ME 570: Robot Motion Planning

Homework 3 Report

By Cameron Cipriano

11/02/2021

**Problem 1: Drawing and Collision Checking for Spheres**

Question 1.1 \_code\_ – Sphere.distance()

This function was just a matter of implementing through np.linalg.norm()

The hollow sphere is the negative of the filled in verison

Question 1.2 \_code\_ – Sphere.distance\_grad()

The gradient of the distance function is . This too was implemented through np.linalg.norm(), taking the negative if the sphere was hollow

Question 1.1 \_optional\_ ­– me570\_hw3.sphere\_testCollision()

Chart, bubble chart

Description automatically generatedChart

Description automatically generated

**Problem 2: The Potential-Based Planner**

Question 2.1 \_code\_ – RepulsiveSphere.eval()

Implemented Equation 3 from the PDF

Question 2.2 \_code\_ – RepulsiveSphere.grad()

Implemented Equation 4 from the PDF utilizing the distance\_gradient function

Question 2.3 \_code\_ – Attractive.eval()

Implemented Equation 1 from the PDF

Question 2.4 \_code\_ – Attractive.grad()

Implemented Equation 2 from the PDF

Question 2.5 \_code\_ – Total.eval()

Implemented equation from the function’s documentation on the assignment utilizing the **eval()** functions for the Attractive and RepulsiveSphere classes.

Question 2.5 \_code\_ – Total.grad()

Implemented equation from the function’s documentation on the assignment utilizing the **grad()** functions for the Attractive and RepulsiveSphere classes.

Question 2.1 \_optional\_ – Using field\_plotThreshold() to Visualize Attractive and Repulsive Potentials

Question 2.6 \_code\_ – Planner.run()

Question 2.7 \_code\_ – Planner.run\_plot()

Question 2.1 \_report\_ – Results from Planner.run\_plot()

Question 2.2 \_report\_ – Visualizing the total potential, , and its gradient

Question 2.3 \_report\_ – Effects of varying repulsive weight and epsilon

Question 2.4 \_report\_ – Relation between the value of toward the final iterations and its effect on the planner’s success/failure.

Question 2.5 \_report\_ – Difference between two goal positions

**Problem 3: CLF-CBF Formulation**

Question 3.1 \_report\_ – Write the expressions for the clover involving and the spade involving

Question 3.1 \_code\_ – me570\_potential.clfcbf\_control()

Question 3.2 \_report\_ – Results from potential\_planner\_runPlot()

Question 3.3 \_report\_ – Results from field\_plotThreshold() to visualize the control for each combination of repulsive weights

Question 3.4 \_report\_ – Trade-off between traditional gradient-based methods and a CLF-CBF formulation

**Problem 4: Jacobian-Based Inverse Kinematics for the Two-Link Manipulator**

Question 4.1 \_report\_ – Write an expression for the Jacobian Matrix such that

Question 4.1 \_code\_ – TwoLink.jacobian\_matrix()

Question 4.1 \_optional\_ – Compare Jtheta\*thetaDot with the results of TwoLink.jacobian(theta, thetaDot)

Question 4.2 \_code\_ – TwoLinkPotential.eval()

Question 4.2 \_code\_ – TwoLinkPotential.grad()

Question 4.3 \_code\_ – TwoLinkPotential.plot\_animate()

Question 4.4 \_code\_ – TwoLinkPotential.run\_plot()

Question 4.1 \_report\_ – Results from twolink\_planner\_runPlot()