Ian Chadwick
11/3/2021

<u>Homework #3 Report</u>

Using 24 Hour Late Pass

Question 2.1: Figures for Conic and Quadratic Potential Based Planner

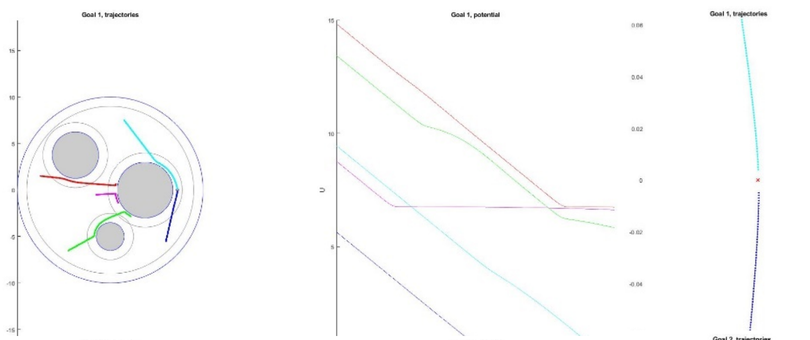


Figure a)
shape = Conic,
NSteps = 10000,
epsilon = 0.001,
repulsive weight = 0.01,
Goal 1 - zoom view (right)

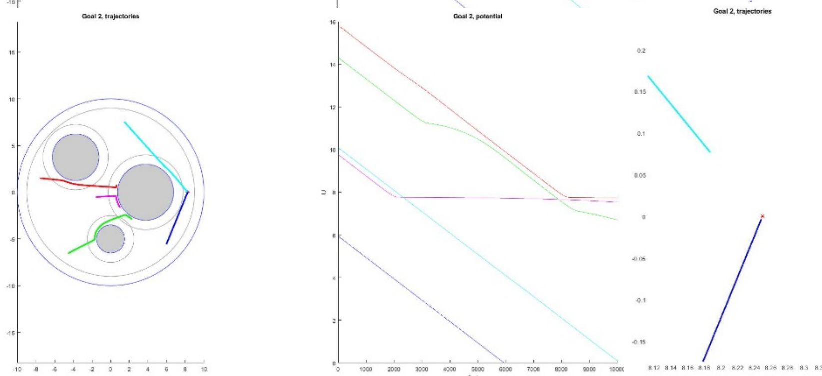


Figure b)
shape = Conic,
NSteps = 10000,
epsilon = 0.001,
repulsive weight = 0.01,
Goal 2- zoom view (right)

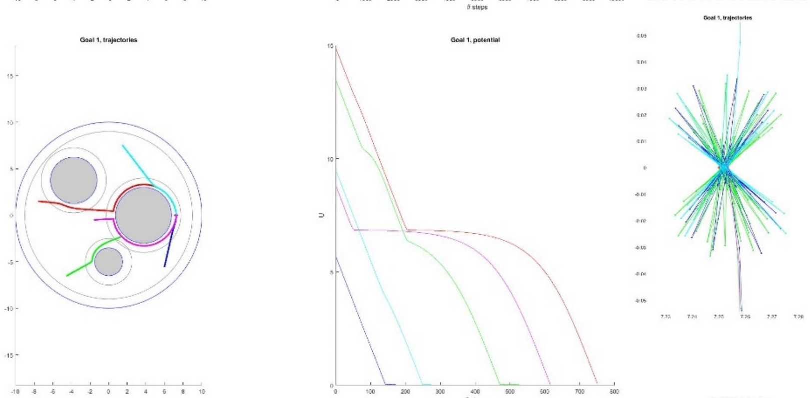


Figure c)
shape = Conic,
NSteps = 750,
epsilon = 0.04,
repulsive weight = 0.02,
Goal 1 - zoom view (right)

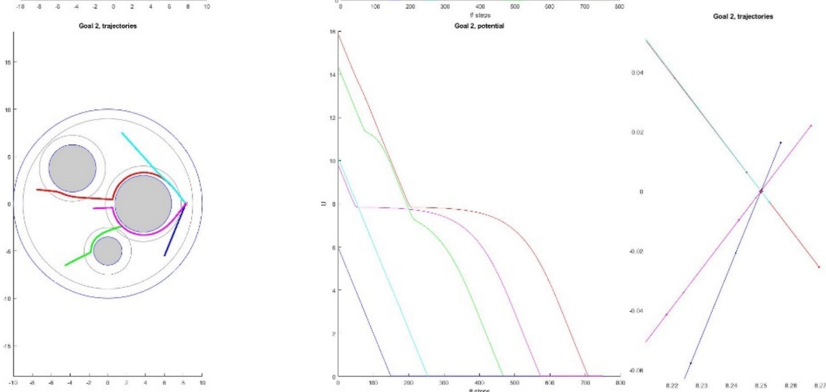


Figure d)
shape = Conic,
NSteps = 750,
epsilon = 0.04,
repulsive weight = 0.02,
Goal 2 - zoom view (right)

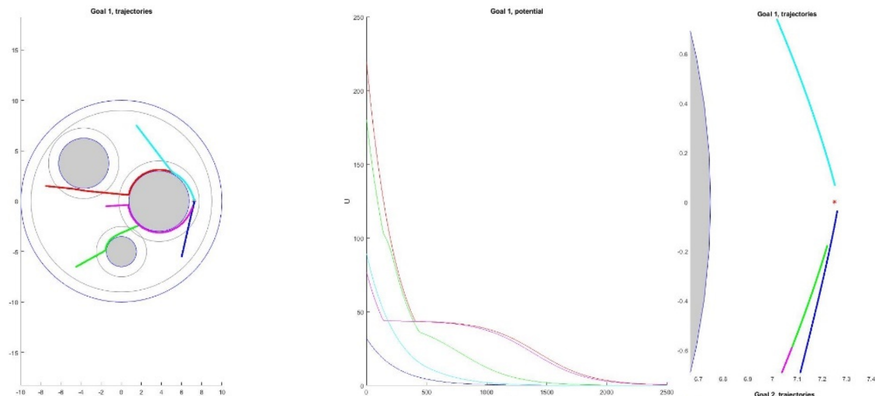


Figure e)
shape = Quadratic,
NSteps = 2500,
epsilon = 0.001,
repulsive weight = 0.01,
Goal 1 - zoom view (right)

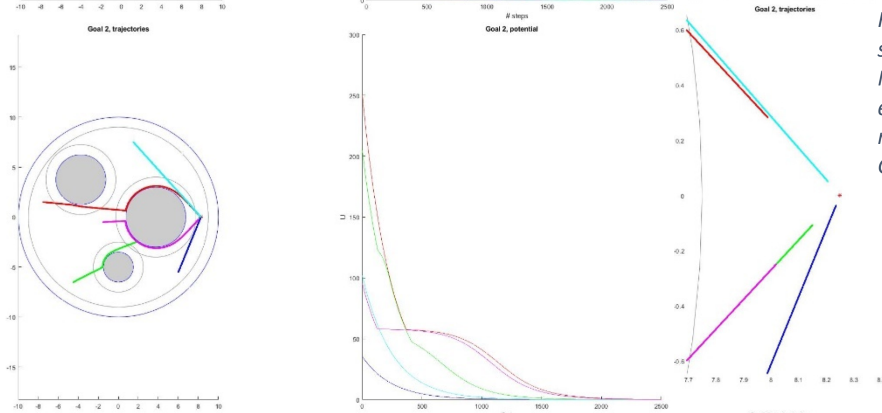


Figure f)
shape = Quadratic,
NSteps = 2500,
epsilon = 0.001,
repulsive weight = 0.01,
Goal 2 - zoom view (right)

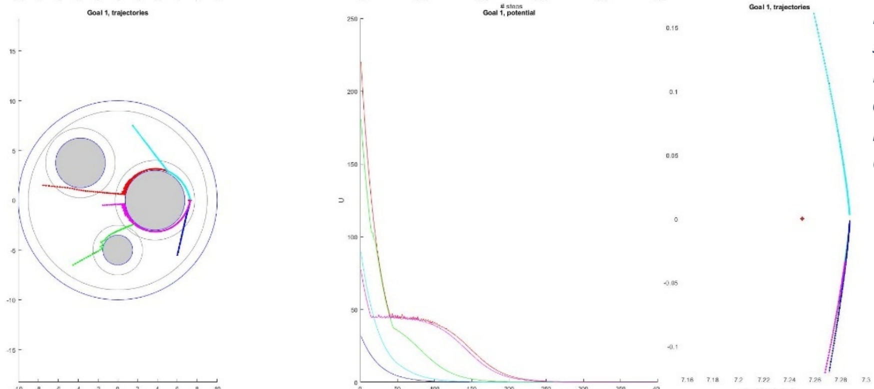


Figure g)
shape = Quadratic,
NSteps = 400,
epsilon = 0.01,
repulsive weight = 0.025,
Goal 1 - zoom view (right)

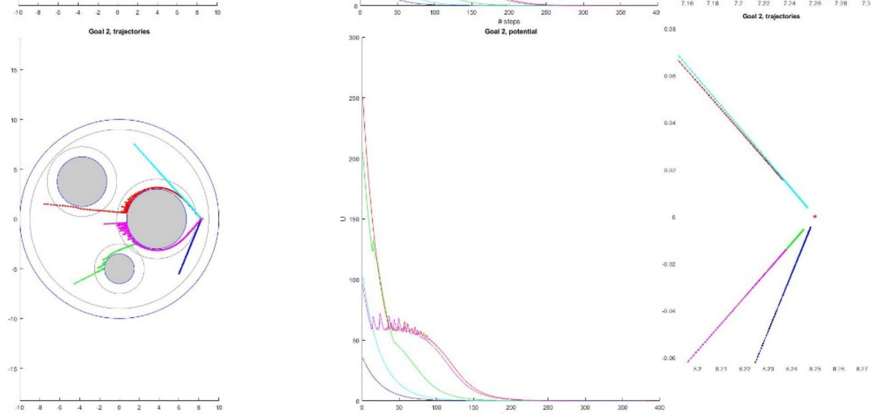


Figure h)
shape = Quadratic,
NSteps = 400,
epsilon = 0.01,
repulsive weight = 0.025,
Goal 2 - zoom view (right)

Question 3.2: Figures for CLF-CBF Formulation

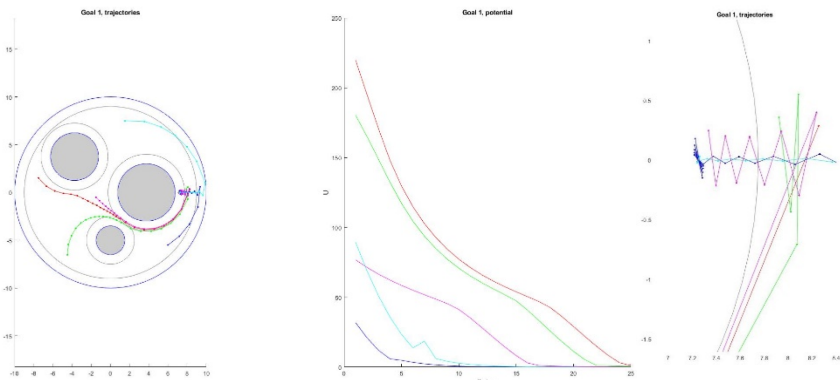


Figure i)
shape = Quadratic,
NSteps = 25,
epsilon = 1.1,
repulsive weight = 0.125,
Goal 1 - zoom view (right)

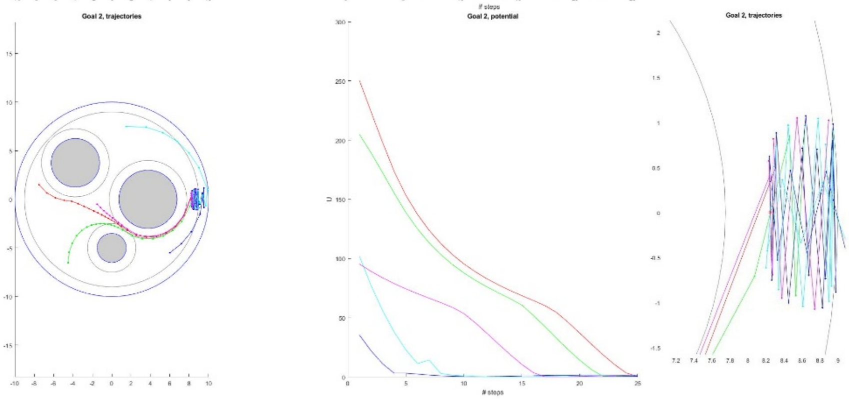


Figure j)
shape = Quadratic,
NSteps = 25,
epsilon = 1.1,
repulsive weight = 0.125,
Goal 2 - zoom view (right)

Question 2.2: Total Potential and Total Potential Gradient Visualization

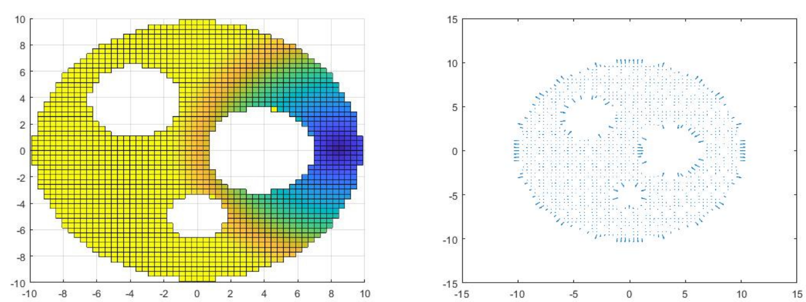


Figure k) Conic, total potential (left) and total potential gradient (right): repulsive weight = 0.01

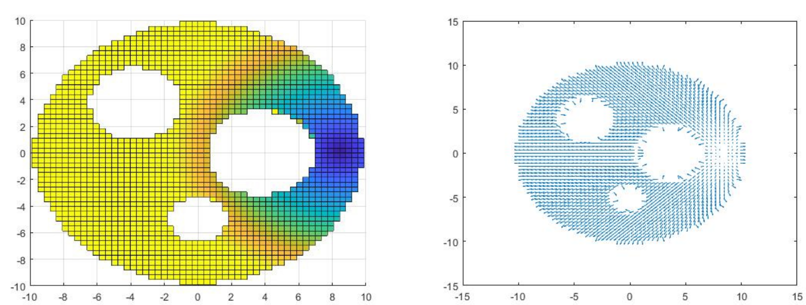


Figure l) Conic, total potential (left) and total potential gradient (right): repulsive weight = 0.025

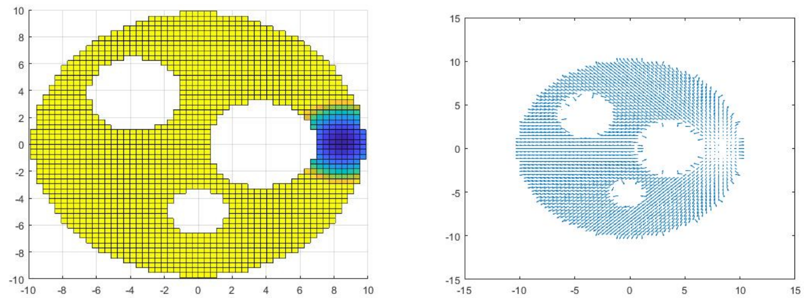


Figure m) Quadratic, total potential (left) and total potential gradient (right): repulsive weight = 0.01

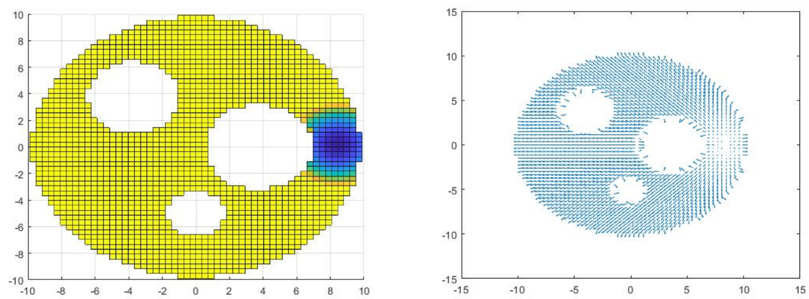


Figure n) Quadratic, total potential (left) and total potential gradient (right): repulsive weight = 0.025

Question 2.3: Effects of Epsilon and Repulsive Weight on Planner Performance

In this planner model, epsilon affects the step size for each of the two types of planners, while the repulsive weight determines how much the obstacles will repel the robot as they enter the distance of influence of each obstacle, following the negative gradient. For each of the planner shapes, the initial parameters tested were the lowest values in the recommended range for both epsilon (0.001) and repulsive weight (0.01). With these starting parameters, in conic planner case (*figures a, b)*, a few of the start locations (*magenta, green and red*) do not reach the goal for these lower values of epsilon, regardless of the value of repulsive weight and would take an inordinate number of steps to reach the goal (>>10000). Two start locations (*teal and blue*) only reach the general vicinity of the goals after many steps (NSteps = 10000).

After some experimentation, epsilon 0.04, repulsive weight 0.025 and NSteps = 750 were used to ensure that the planner successfully found a path for all 5 starting locations (*figures c and d*). However, in figure c, once the robots do reach the vicinity of the goal, the chattering effect can be observed as the planner causes the robots to bounce back and forth over the goal. This occurs because the step size in a conic planner is fixed, and it is too large to allow the robots to successfully be within the threshold of "reaching" the goal without overshooting it. Since the goal is also within the influence distance of the obstacle, they are affected by the repulsive gradient. This causes the robot to step back and forth towards and away from the obstacle over the goal until the maximum number of steps is reached.

For the same low starting values of epsilon (0.001) and repulsive potential (0.01), the quadratic planner (*figures e and f*), it runs into an issue where even after 2500 steps, it never technically reaches the goal. This is due to the fact that the epsilon value is too small and because for a quadratic planner, it takes large steps initially, but the step size is reduced as the robot gets closer to the goal. However, it allows all 5 starting locations for the robot to approach relatively close to the goal with a much smaller epsilon and Nstep (2500) value than the conic.

After experimentation with different parameter values, it was found that the largest epsilon value in the suggested range (0.01) and a repulsive weight 0.025 allowed all the starting locations to reach the goal within 400 steps.

In general the behavior of the planner for these scenarios for values of epsilon and repulsive weight were:

- **Small epsilon/Small repulsive weight**:
  - For the conic, this was the case where only the robots from some of the starting locations were able to reach the goal location, even after many steps. The step size was too small and in some cases the robot was essentially stuck as the potential essentially "flattened out".
  - For the quadratic, this was the case where it took a large amount steps for all of the robot starting locations to reach the general vicinity of the goal.
- **Small epsilon/Large repulsive weight**:
  - For the conic in this case, the same starting locations that did not reach the goal in the previous case still did not reach the goal, and it took longer for the other starting locations to reach the goal as the stronger repulsive potential pushed the robot further away from the object increasing the total distance traveled. For extremely large repulsive weights, the potential curve for some start locations would flatten out, and the robot would get "stuck" indicating a local minimum.
  - For the quadratic case, it behaved very similarly, where the large repulsive potential from the obstacles made the robots take a longer path, and which caused some starting locations not to reach the goal. There was also a similar problem where the potential curve flattened out and the robot became stuck in a local minimum.
- **Large epsilon/Small repulsive weight**: This case caused some issues with the both planners, but with the quadratic planner for several of the starting locations (*magenta, red and green*), the planner would step too far toward the object, because it was repulsive potential was too weak. This would cause the robot to either get stuck in/on the obstacle and there were some occasional errors where the robot appeared to be "launched" outside of the configuration space entirely. I suppose this is because the robot was able to get too close to the object where the value of the repulsive potential became extremely large. It behaved similarly for the conic planner, but in this case it only got stuck inside the obstacles and no errors where it ended up outside the world occurred.
- **Large epsilon/Large repulsive weight**: This generally had similar performance to the previous large epsilon case, where starting locations on the opposite side of the obstacles from the goal ended up stepping too far into the obstacle and becoming stuck. This happened for both the conic and the quadratic cases.

These observations are consistent with what we discussed in class. If the epsilon is too low, it won't reach the goal, or if it is too high, it runs the risk of stepping too far into obstacles. Similarly, if the repulsive potential is too low, it will allow the planner to step into the obstacle and if it is too high it can cause the robot to get stuck since the repulsive potential can cause the robot to be stuck in a local minimum as the repulsive potential "cancels out" the attractive potential.

Question 2.4: Relation Between Value of Potential U and Success/Failure to Reach Goal

The planner was able to guide the robot successfully to the goal in the cases when the potential, ***U*** was able to successfully reach zero, or get close to zero. Generally, if the slope of the potential curve was decreasing fast enough to reach zero in the number of steps, it was able to successfully reach the goal location. If the potential curve was not decreasing fast enough to reach zero in Nsteps, it failed to reach the goal, but it may reach the goal if provided sufficiently more steps.

However, if the robot got stuck, the potential curve tended to show a relatively flat line from the point it got stuck to the cutoff time Nsteps and the robot failed to reach the goal entirely (*figure a and b, red and magenta)*. In this case, it is unclear if more steps would allow it to reach the goal and it would depend on how much the curve was decreasing, if at all. In the other cases, when the robot ended up either inside of an obstacle or jettisoned outside of the sphere world, the potential curve ended abruptly where the error occurred, and it also failed to reach the goal.

Question 2.5: Difference Between Goals and Effect of Shape

The difference between the goals provided was the fact that **Goal 1** was inside of the influence distance of one of the obstacles, whereas **Goal 2** was outside of the obstacle's influence distance. This had an effect on the trajectory of the robot as it approached the goal. In some cases, (*figure g, goal 1*) it meant that the repulsive potential from the object was too strong for the robot to actually "reach" the goal point. Instead it would get within the range allowed by the repulsive potential. Or, in other cases, (*figure c, goal 1*) it caused the robot to chatter away, and then toward the goal location as the repulsive potential pushed it away from the obstacle. In the case of **Goal 2,** if the robot was able to reach **Goal 2** it usually stopped at the goal location. The exception is the conic case, where a less dramatic chattering effect would occur as the fixed step size of the planner caused the robot to repeatedly overstep the goal (*figure d*).

Question 3.1: Expression for $u^*(x)$ in CLF-CBF Formulation

The value for $\clubsuit$ = **uRef** is:

$$\clubsuit = -\nabla U_{attr}$$

The value for $\spadesuit$ is:

$$\spadesuit = A_{barrier} u + b_{barrier} \leq 0$$

Where:

$$A_{barrier} = -\nabla h^T$$
$$b_{barrier} = -\alpha h$$

And: α = repulsive potential, $h = d_i$, $c_h = \alpha h$

$$\spadesuit = -\nabla h^T u - c_h \leq 0 \quad \text{or} \quad \spadesuit = -\nabla {d_i}^T u - \alpha d_i$$

Question 3.3: Control Field $u^*(x)$ Visualization for CLF-CBF Formulation

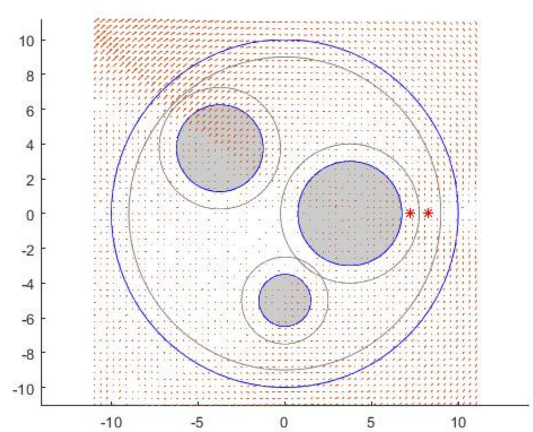


Figure m) Quadratic, for repulsive field = 0.125

Question 3.4: Trade-off Between Gradient Based Method and CLF-CBF Formulation

One of the largest differences between the two is the number of steps that it takes to reach the end goal. For the CLF-CBF formulation, it is able to utilize a much higher epsilon value, which means that it is able to get to the goal in significantly less steps since it is subject to the CBF constraints. These constraints ensure that the planner does not overstep into the obstacle, which is the reason that traditional gradient based methods can't use large values of epsilon. However, the computation time for the CLF-CBF formulation was significantly longer and it would likely need to be computed offline and couldn't be run locally on limited hardware. Additionally, the gradient based method seemed to be able to get closer to the goal, especially as it approached the goal, when compared to the CLF-CBF method, which ended up "zig-zagging" as it approached the goal.

Question 4.1: Expression for Jacobian Matrix

Let the velocity of the end effector point be:

$$\dot{\boldsymbol{X}} = \frac{d}{dt}({}^{W}\boldsymbol{p}_{eff})$$

And the angular velocities of the joints be:

$$\dot{\boldsymbol{\theta}} = \begin{matrix} \dot{\boldsymbol{\theta}}_1 \\ \dot{\boldsymbol{\theta}}_2 \end{matrix}$$

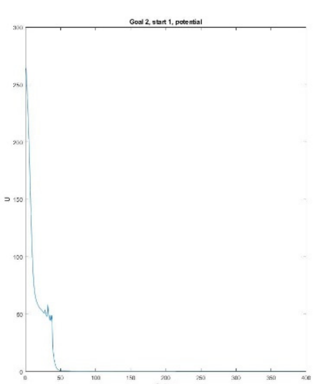
Since the length of each link was given as **5,** we can write the equation for the Jacobian below:

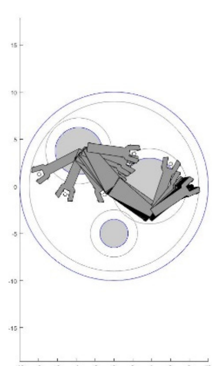
$$\mathbf{J(q)} = \begin{bmatrix} -5sin\theta_1 - 5sin(\theta_1+\theta_2) & -5sin(\theta_1+\theta_2) \\ 5cos\theta_1 - 5cos(\theta_1+\theta_2) & 5cos(\theta_1+\theta_2) \end{bmatrix}$$

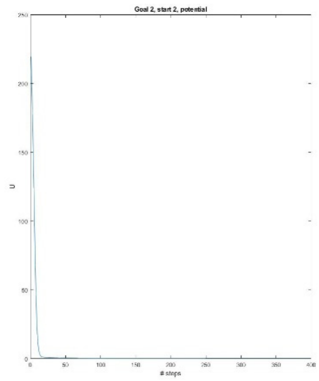
And the velocity of the end effector can be calculated below by multiplying angular velocity by the Jacobian:

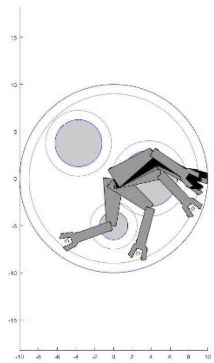
$$\dot{\boldsymbol{X}} = \mathbf{J(q)}\,\dot{\boldsymbol{\theta}} = \begin{bmatrix} -5sin\theta_1 - 5sin(\theta_1+\theta_2) & -5sin(\theta_1+\theta_2) \\ 5cos\theta_1 + 5cos(\theta_1+\theta_2) & 5cos(\theta_1+\theta_2) \end{bmatrix} \bullet \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

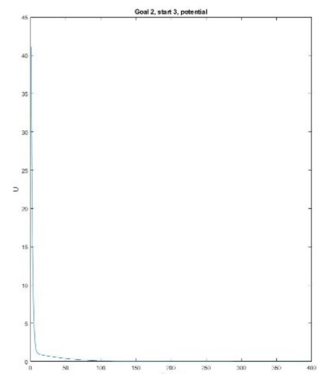
Question 4.2: Figures for Two-link Manipulator End Effector Path Planning

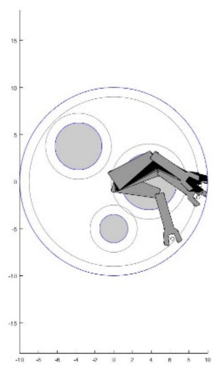


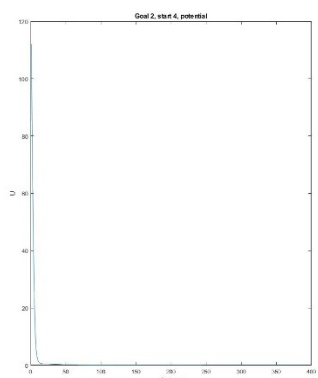


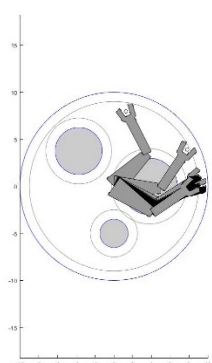

Goal 2, start 2, potential
U
steps

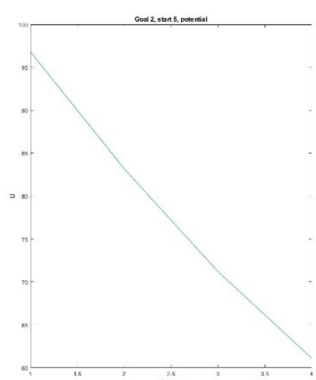



Goal 2, start 3, potential
U
steps




Goal 2, start 4, potential
U
steps




Goal 2, start 5, potential
U
steps