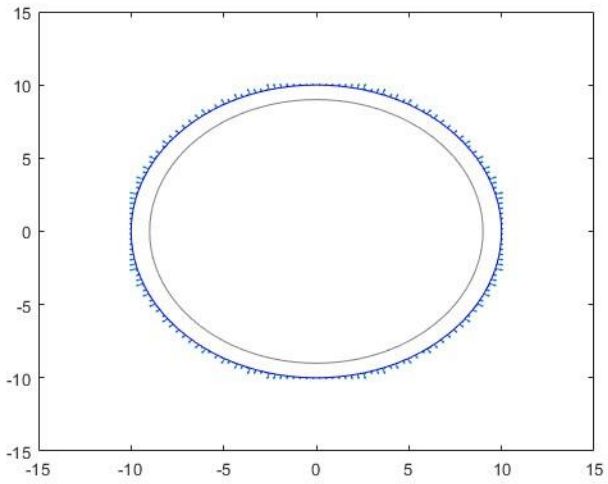
HomeWork 3 Report

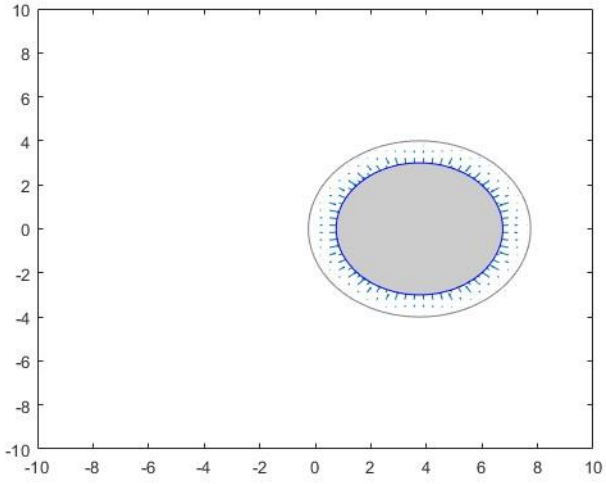
By

U25712111(Adwait Kulkarni)

Question Report 2.1 : Use the provided function grid_plotThreshold ( ) to visualize potential_repulsiveSphereGrad( ) for the first two spheres in the sphere world, and overlap the plot of the sphere in each plot; use a different figure for each sphere. Include the figures in your report.

Solution :





Question Report 2.2 : In the figures generated with the previous question, should the arrows point toward or away from the obstacle? Why? Why are the arrow only visible around the contour of the obstacle?

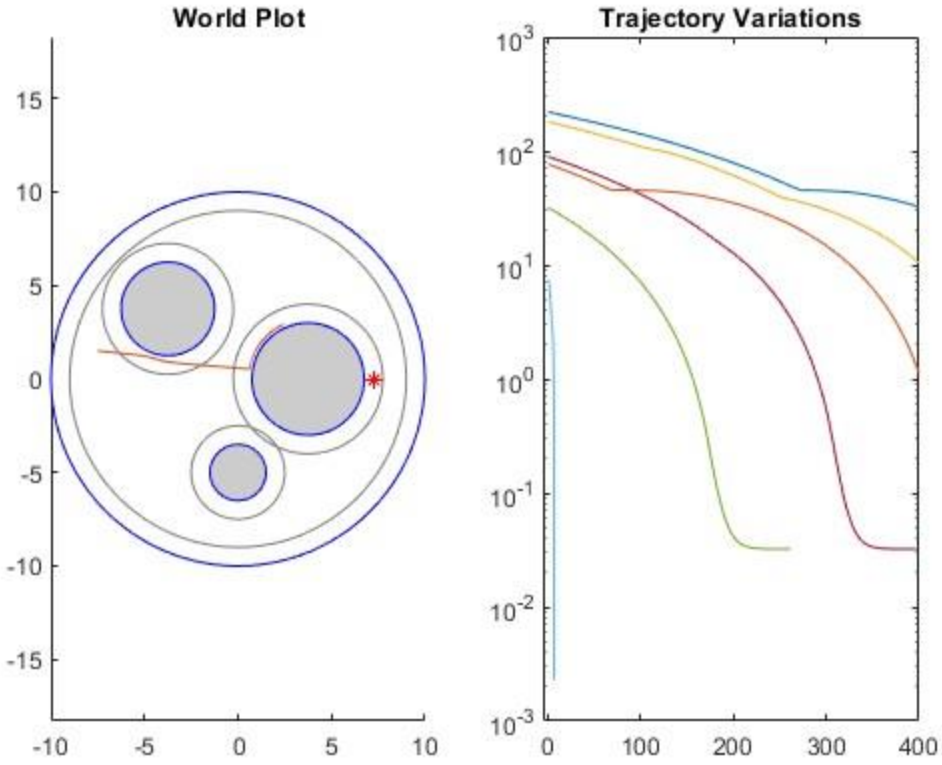
Solution :

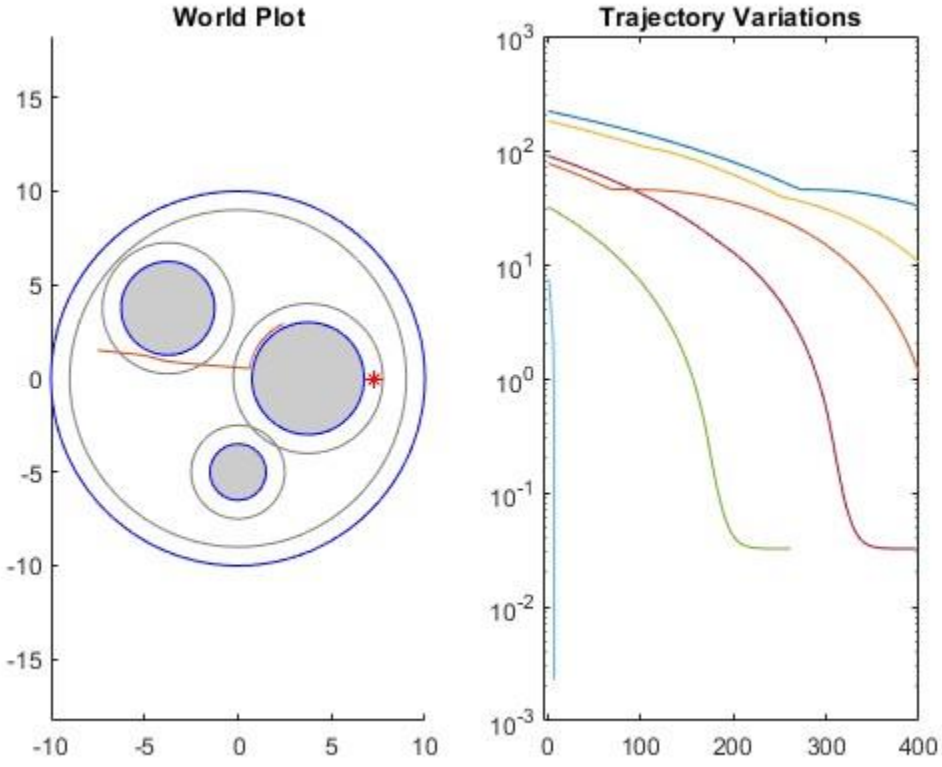
Direction of the Arrows: For the direction of arrows they should point away from the obstacle. This is because the repulsive potential increases as the robot or agent gets closer to the obstacle. The arrows represent the gradient of this potential which points in the direction of the steepest increase of the potential. Now as Since the potential is repulsive and meant to push the robot away from the obstacle the arrows should point outward away from the obstacle.

Visibility Around the Contour: The arrows are only visible around the contour of the obstacle because of the nature of the repulsive potential. Typically, the repulsive potential is designed to be non-zero only within a certain distance from the obstacle which is known as distance of influence(Also used in the coding part of this assignment). Beyond this distance, the repulsive potential is zero implying there's no force acting on the robot to push it away. This is done to ensure that the robot is only influenced by obstacles that are in close proximity. Hence, the arrows are only visible within this zone of influence around the obstacle. (One of the other reasons is it is done for computational efficiency so we don't compute repulsive forces everywhere but the primary reason is mentioned above)

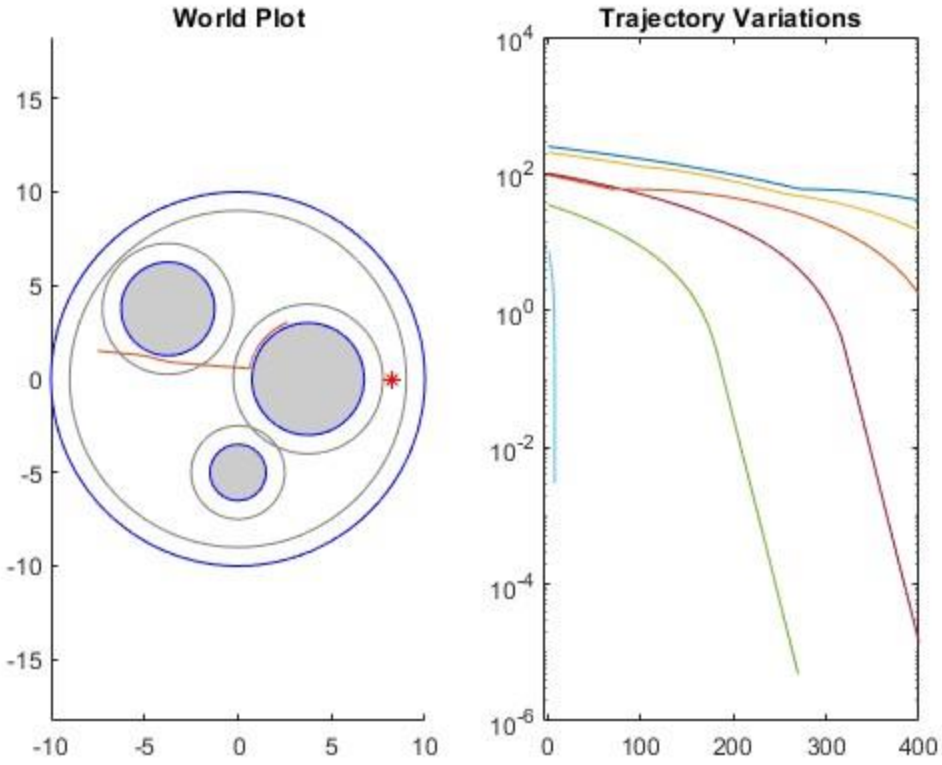
Question Report 2.3 : Implement a function that runs the planner on the provided data, and visualizes the results to explore and illustrate the effect of the different parameters in the planner. This question will generate many figures. Include all figures in your report, trying to have all of them organized in one or two pages, for ease of comparison.

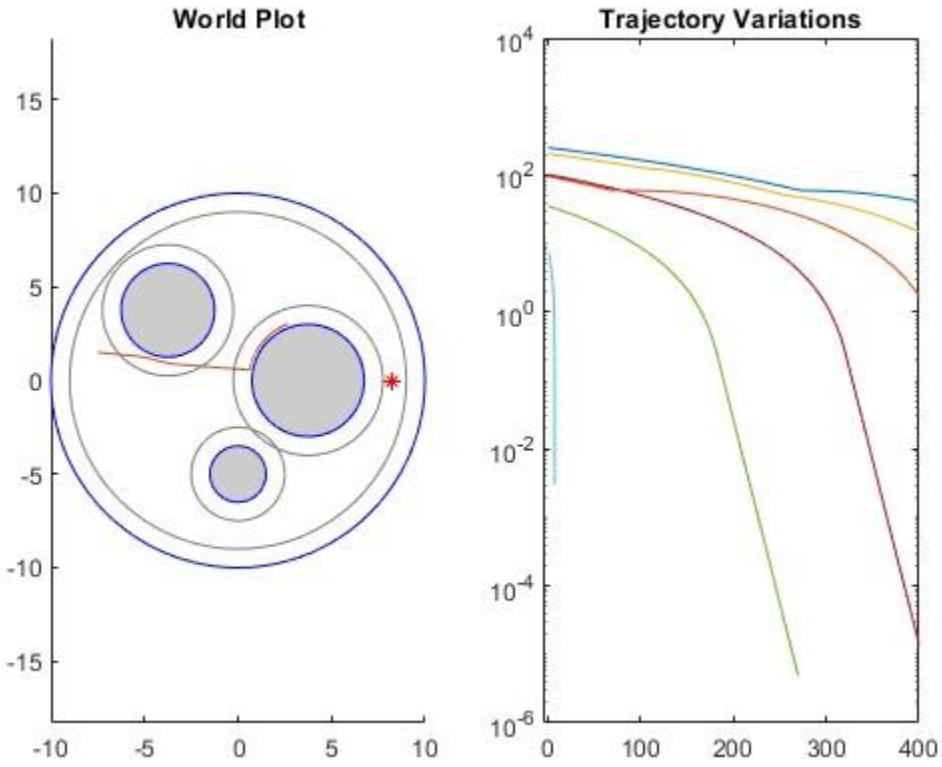
Solution :





When Nsteps is 100 and Replusive Weight is 0.1 and Epsilon is 1e-2





When Nsteps is 200 and Replusive Weight is 0.1 and Epsilon is 1e-2

NSteps(plannerParameters.NSteps):

- Influence:
    - A smaller value might result in the planner stopping before it finds an optimal path or even any path at all, especially if the start and goal locations are far apart or the world is densely populated with obstacles.
    - A larger value allows the planner to explore more and potentially find a better path but at the cost of more computation time.

RepulsiveWeight(potential.repulsiveWeight):

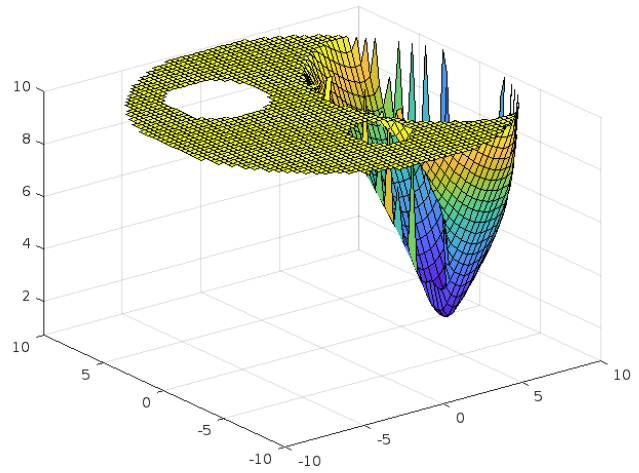
- Influence:
    - A smaller repulsive weight might cause the planner to generate paths that come very close to or even intersect obstacles since the repulsion from obstacles is weak.
    - A larger repulsive weight will push the trajectory away from obstacles, making it safer but possibly longer or more circuitous.

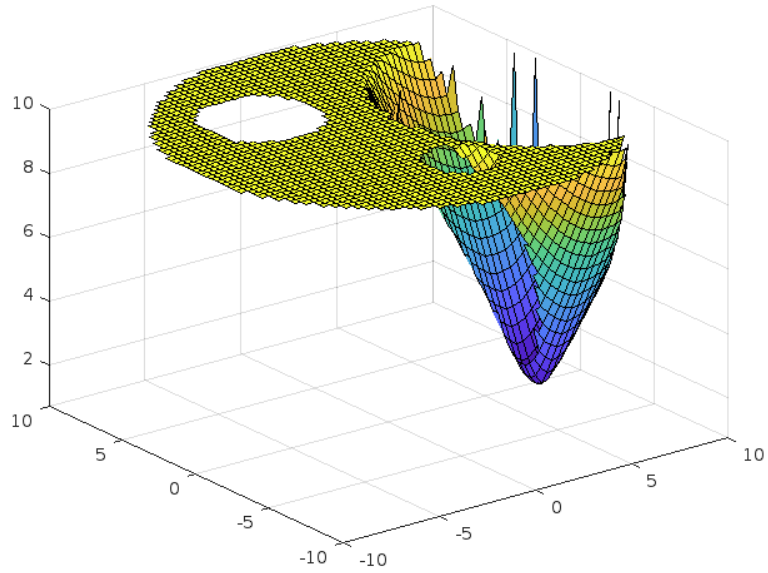
Epsilon(plannerParameters.epsilon):

- Influence:
    - A smaller epsilon value would make the planner take smaller steps. This can result in finer paths, which are more accurate but may take longer to compute.
    - A larger epsilon value would make the planner take bigger steps. This can speed up the planning process, but the resulting paths might not be as refined or accurate, and there's a risk of skipping over optimal paths or getting stuck in local minima.

Question Report 2.4 : Use the function grid_plotThreshold( ) to visualize the total potential U, and its gradient ∇U in two separate figures for each one of the combinations repulsiveWeight and shape included in the previous question (report 2.3) (for this question, it is sufficient to consider only one of the two goals). Include the images in you report.

Solution :





First Image is for Quadratic and Second Image is for Conic

Question Report 2.5 : Comment on the effects of varying each one of the parameters repulsiveWeight and epsilon . Explain why the planner behaves in the way you see, explicitly explaining what happens for the four cases where the two parameters are, respectively, small/small, small/large, large/small, and large/large. Additionally, comment on whether the results you see in practice are consistent with what discussed in class.

Solution : The planner behaves the way we see for those four cases where the parameters vary as small and large permutation with each other. All the 4 cases reason are explained below:

Small /Small : In this case the attractive force is weak, so the robot might not be strongly pulled toward the goal. Since epsilon is also small the repulsive forces from obstacles will be sharp and might cause the robot to get stuck in local minima because the influence of obstacles is very localized.
Small /Large : In this case the robot will be weakly attracted to the goal. However, the repulsive forces are now smoothed out due to a larger epsilon. This means that the influence of obstacles is felt from a greater distance but it is more gradual. The robot might be able to navigate around obstacles more easily but might also be easily diverted from its path to the goal.

Large /Small : In this case the robot will be strongly attracted to the goal. The repulsive forces will remain sharp, while the strong attractive force might help the robot overcome some local minima, it might also cause it to try to push through narrow passages or even attempt to go directly through obstacles because of the strong pull towards the goal.

Large /Large : In this case the robot will be strongly attracted to the goal and the repulsive fields would be smoothed out. This might be the most balanced scenario, as the robot would be strongly motivated

to reach the goal but would also have a smoothed-out understanding of obstacles, helping it navigate the environment more efficiently.
Additionally the results that I saw were not entirely consistent with what was discussed in the lectures, I might say like 65-70% consistent with was discussed in the class.

Question Report 2.6 : Comment on the relation between the value of the potential U toward the end of the iterations, versus the fact that the planner correctly succeeded or failed. Explain different reasons why the planner might fail, and why changing the various parameters can improve the situation.

Solution :
Potential U and Planner's Outcome:

- Low Value of U: If the potential U is consistently reducing and stabilizes towards a minimal value by the end of iterations, it is an indication that the planner is moving towards the goal or desired state without major hindrances.
- High Value or Oscillating U: A persistently high potential or oscillations in U suggests the planner is either encountering obstacles, stuck in local minima, or oscillating between potential fields.

Reasons for Planner Failure:

- Local Minima: Peaks in the potential field, like those seen in the images, can lead to local minima. If the planner gets trapped in these, it might not find a path to the goal.
- Oscillations: If the planner is caught between strong repulsive and attractive potentials, it might oscillate and not make progress towards the goal.
- Narrow Passages: Tight spaces or narrow passages surrounded by high potentials can be challenging. The repulsive forces might prevent the planner from passing through.

Improving Planner Success with Parameter Tuning:

- Adjusting Weights: Balancing the weights of attractive and repulsive potentials can influence the planner's trajectory. Too much repulsion can prevent the planner from approaching the goal, while too much attraction can make the planner run into obstacles.
- Grid Resolution: A finer grid can capture more details, which can be useful in complex environments. However, it's also computationally more intensive.
- Smoothing: Applying a smoothing algorithm can help the planner navigate around local minima and reduce oscillations, but oversmoothing might cause the planner to miss narrow passages.
- Dynamic Parameter Adjustment: Parameters can be adjusted dynamically based on the situation. For instance, if the planner is stuck, it can momentarily increase the attractive potential.

Question Report 2.7 : What is the difference between the two goals included in the provided dataset? In relation to this, what is the effect of the parameter shape ?

Solution : For the case where there are 2 goals provided in the dataset I saw not much of a difference like they have a computational or numerical difference of 1 units between them, and also in the output and the path created for both of those were nearly similar. The shape i.e either conic or quadratic had a profound effect, due to its proximity to the obstruction, the potential has a disproportionately large

impact on one of the target sites. The alternative potential objective site is further from the obstruction and so less affected by it.

Question Report 3.1 : In the report, write the expressions for ♣ (involving Uattr) and ♠ (involving di(x)). For this and the questions below, denote the constant appearing in the constraint (5b) as C_h. Please pay attention to the directions of the inequalities.

Solution:
Firstly,
• U_{attr} serves as the Control Lyapunov Function (CLF).
• di(x) acts as the Control Barrier Function (CBF) for every barrier.

From the outlined QP structure: u∗(x) = argmin u‖u− ♣ ‖^2, subject to the condition: ♠ ≤ 0.

The term ♣ essentially captures the gap between the actual control u and the ideal control that minimizes U_{attr}. In CLF-based control approaches, the optimal control udes is commonly derived from the gradient of U_{attr} in relation to state x. As a result, ♣ is the inverse gradient of U_{attr} relative to x:

♣ = −∇Uattr(x).

As for ♠, being dependent on di(x) (distance to the i-th barrier), a standard approach for the CBF constraint is to ensure the robot maintains a certain distance from the barrier. Thus, the C_h value for the i-th barrier can be defined as: $C_{hi} = C_h$ - di(x), where C_h represents a designated safety distance from the barrier. This safe distance refers to desired minimum distance from obstacle and it is also know as distanceInfluence.(We used this in the code part for this assignment)

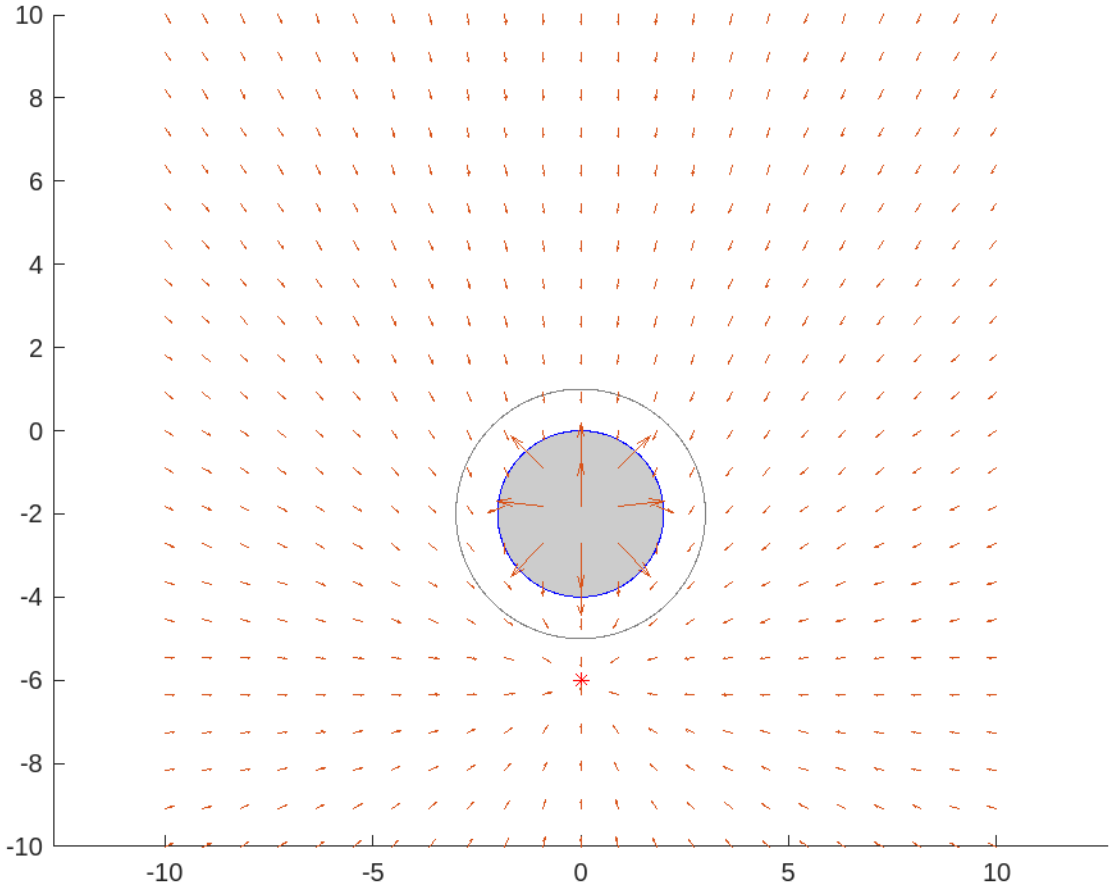
Question Report 3.2 : Comparing (6) with (5), and knowing that there are 4 obstacles in the environment, give the dimensions of $A_{barrier}$ and $B_{barrier}$.

Solution :

For each obstacle, there is a constraint so since there are 4obstacles, we will have 4 constraints. Also the control *u* has 2 components. Therefore:
For $A_{barrier}$: Which multiplies with control u has 4 rows for each obstacle and 2 columns for each component of u . Hence the size of $A_{barrier}$ is 4 X 2.
For $B_{barrier}$: It has a single value for each obstacle, hence making its size 4 X 1.

Question Report 3.3 : Run the function from provided 3.2 above, and include the produced image in the report. You should see that the field induces a flow that goes around and has a stable equilibrium around the point [0]
[−6].

Solution :

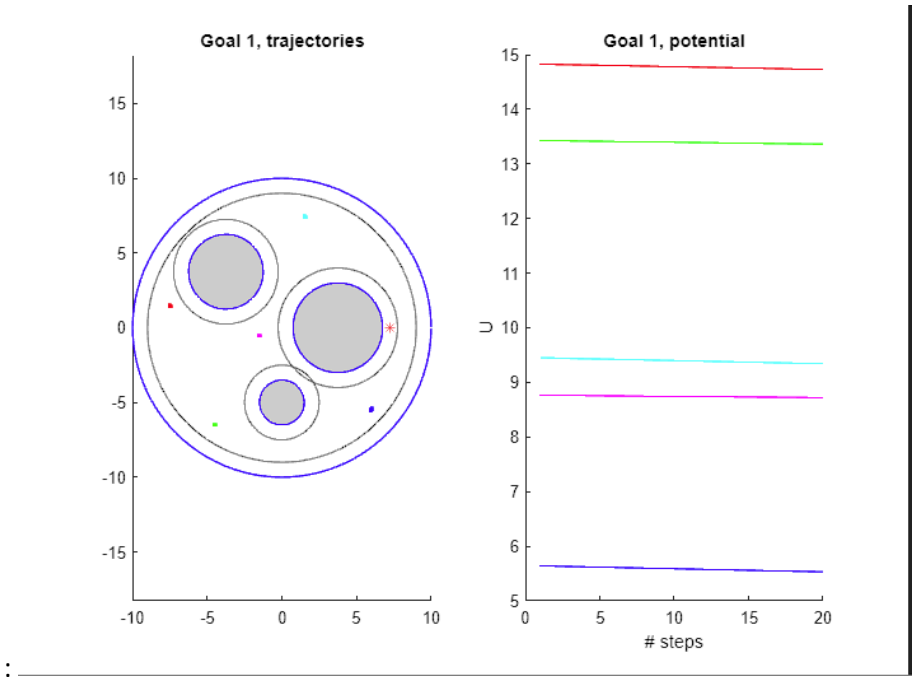


Question Report 3.4 : Explain why the field is essentially zero at the origin (i.e., at the top of the edge of the sphere). To get full points, your explanation should be based on (5) and what we have seen in class.

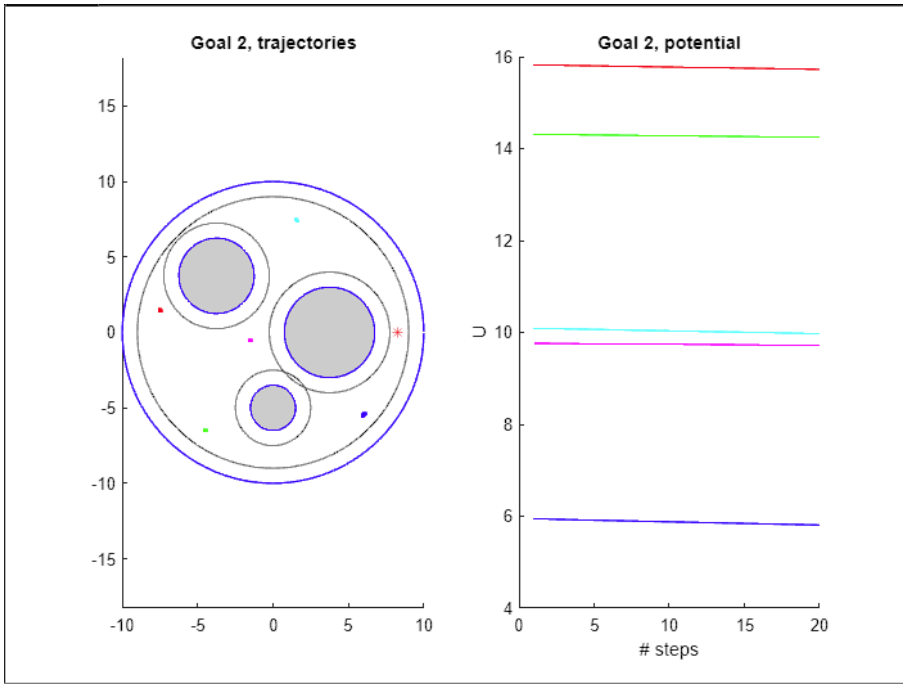
Solution : **T**he near-zero field at the origin (top of the sphere's edge) is due to the interplay between the desire to be close to a certain point/value ♣ and the hard constraint of not entering the sphere. This balance, when optimized, results in a near-zero motion at that specific point on the sphere. (This is the best I could explain based on equation 5 and somewhat I remember from class)

Question Report 3.5 : This question is analogous to report 2.3, but using the CLF-CBF framework instead of the gradient of the potential.

Solution



:



Question Report 3.6 : Use the function grid_plotThreshold ( ) to visualize the control field u∗(x) for the value of repulsiveWeight included in the previous question (report 2.3). Make sure to superimpose the field on top of the world map. Note that the computation of u∗(x) will take significantly longer than just the gradient, hence you might want to reduce the size of the grid passed to grid_plotThreshold ( ) (e.g., use a 10 × 10 grid). Include the images in your report.

Solution :

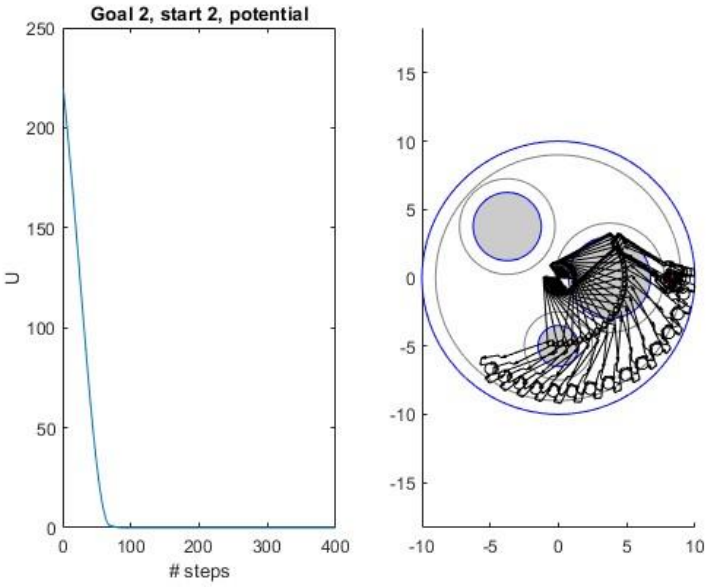
Question Report 3.7 : Comment on the trade-off between traditional gradient-based methods and a CLF-CBF formulation.

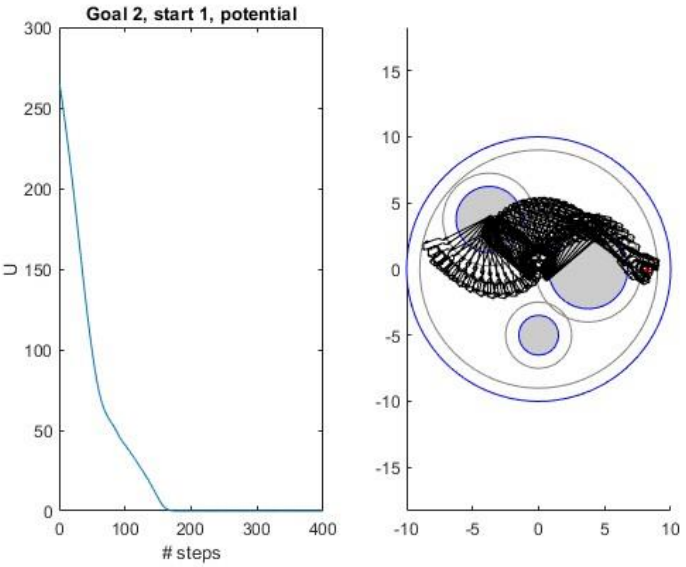
Solution :

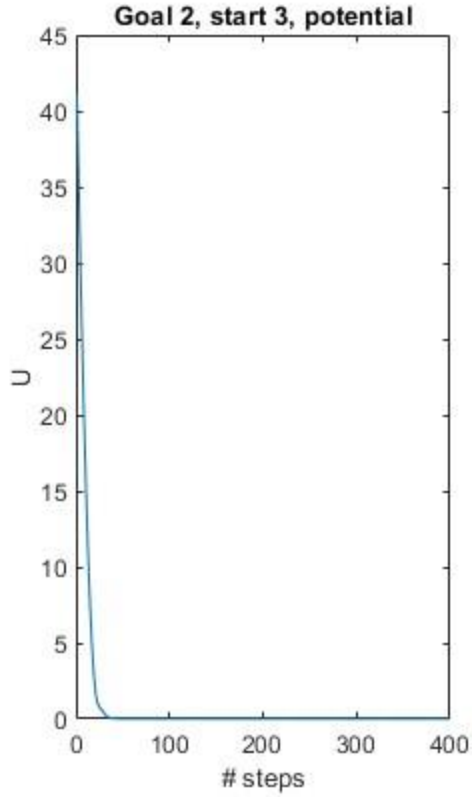
- If safety and stability are important then a CLF-CBF formulation might be preferred despite its complexity. In contrast if you are looking for a quick solution and the problem landscape is relatively simple gradient-based methods might suffice.
- Gradient-based methods often rely on hyperparameters like step size, which require careful tuning. Improper tuning can lead to divergence or very slow convergence. On the otherhand while CLF-CBF also has design parameters their main and primary functions are to ensure safety and performance. This can make CLF-CBF more robust to uncertainties or disturbances in the system especially when the CLFs and CBFs are designed appropriately.
- For systems with complex nonlinear dynamics and hard constraints CLF-CBF might offer better performance, while simpler well-behaved problems might be easily tackled with gradient-based methods.

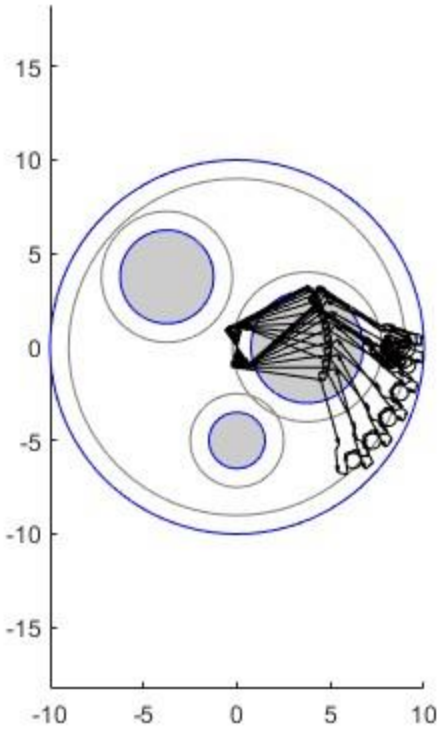
Question Report 4.2 : Show the results of twolink_planner_runPlot ( ) for (one) combination of repulsiveWeight and epsilon that makes the planner work reliably. For the argument plannerParameters.NSteps , use 400 , and for potential.shape , use 'quadratic' . Note that plannerParameters.U , and plannerParameters.control are set by the function, so they do not need to be set in the argument. In your report, try to have all figures on the same one or two pages for ease of comparison. If it is too difficult to get the planner work reliably for all goal configurations, it is fine to include the results using different parameters for different goals, or just for a few out of the five goals. Note that, in this problem, we are considering only collisions between the spheres and the end effector of the manipulator; we are not considering collisions between the spheres and the links of the manipulator; in other words, it is normal to have overlap between the manipulator and the spheres, as long as the end effector is not inside an obstacle.

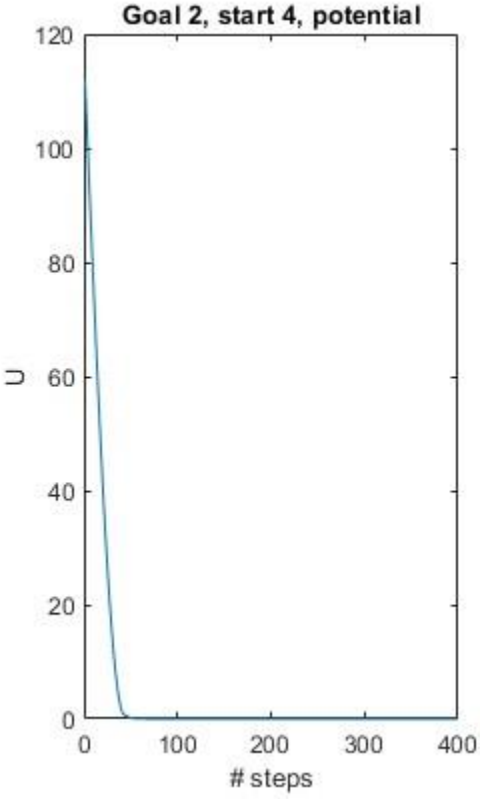
Solution :

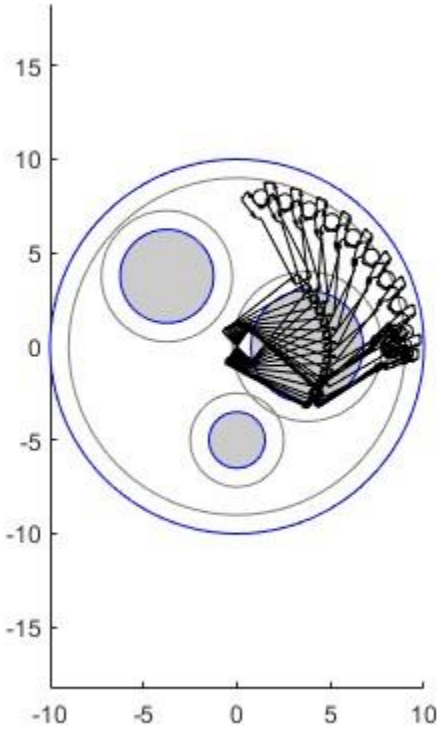

Goal 2, start 2, potential
U
steps
250
200
150
100
50
0
0
100
200
300
400
15
10
5
0
-5
-10
-15
-10
-5
0
5
10

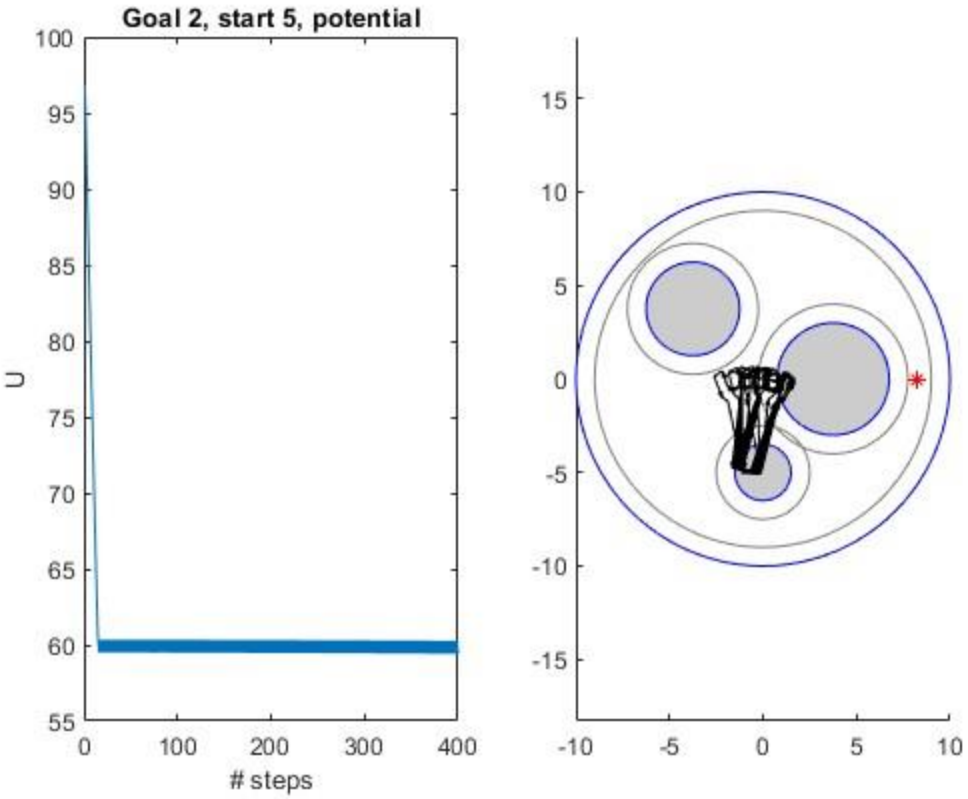

Goal 2, start 1, potential
U
steps
300
250
200
150
100
50
0
0
100
200
300
400
15
10
5
0
-5
-10
-15
-10
-5
0
5
10


Goal 2, start 3, potential
U
steps
0
5
10
15
20
25
30
35
40
45
0
100
200
300
400


15
10
5
0
-5
-10
-15
-10
-5
0
5
10


Goal 2, start 4, potential
U
steps
0
20
40
60
80
100
120
0
100
200
300
400


15
10
5
0
-5
-10
-15
-10
-5
0
5
10



Epsilon for these images is 3e-2 and Repulsive Weight is 0.05 and the Nsteps are 400