MEAM 520

Lab 5: Potential Field

Xiaozhou Zhang / Xinlong Zheng

xzzhang@seas.upenn.edu

xinlongz@seas.upenn.edu

1 Method

1.1 Attractive field

In general, the potential field U for each origin o_i of the DH frame is an additive field consisting of one component that attracts the robot to q_{f} and a second component that repels the robot from the obstacle (SHV p169), as

$$U_i(q) = U_{\mathrm{att},i}(q) + U_{\mathrm{rep},i}(q)$$

In terms of attractive potential field $U_{\mathrm{att},i}$ for o_i, we choose to combine the quadratic conic potential, so that the conic potential attracts o_i when it is distant from its goal position, and the quadratic potential attracts o_i when it is near its goal position. Such field (SHV p170) is given by

$$U_{\mathrm{att},i}(q) = U_{\mathrm{rep},i}(q) = \begin{cases} \frac{1}{2}\zeta_i \|o_i(q) - o_i(q_{\mathrm{f}})\|^2 & ; \quad \|o_i(q) - o_i(q_{\mathrm{f}})\| \le d \\ \zeta_i \|o_i(q) - o_i(q_{\mathrm{f}})\| - \frac{1}{2}\zeta_i d^2 & ; \quad \|o_i(q) - o_i(q_{\mathrm{f}})\| > d \end{cases}$$

in which ζ_i is a parameter used to scale the effects of the attractive potential, d is the distance defines the transition. The workspace attractive force for o_i is equal to the negative gradient of $U_{\mathrm{att},i}$, given as

$$F_{\mathrm{att},i}(q) = -\nabla U_{\mathrm{att},i}(q) = \begin{cases} -\zeta_i (o_i(q) - o_i(q_{\mathrm{f}})\,) & ; \quad \|o_i(q) - o_i(q_{\mathrm{f}})\| \le d \\ -d\zeta_i \dfrac{\big(o_i(q) - o_i(q_{\mathrm{f}})\big)}{\|o_i(q) - o_i(q_{\mathrm{f}})\|} & ; \quad \|o_i(q) - o_i(q_{\mathrm{f}})\| > d \end{cases}$$

1.2 Repulsive field

In terms of repulsive field $U_{\mathrm{rep},i}$ for o_i, we want it to repel the robot from the obstacles and shall exert little influence when the robot is far away from obstacles. We define ρ_0 to be the distance of influence of an obstacle. Our potential function is given by

$$U_{\mathrm{rep},i}(q) = \begin{cases} \frac{1}{2}\eta_i \left(\dfrac{1}{\rho\big(o_i(q)\big)} - \dfrac{1}{\rho_0}\right) & ; \quad \rho\big(o_i(q)\big) \le \rho_0 \\ 0 & ; \quad \rho\big(o_i(q)\big) > \rho_0 \end{cases}$$

in which η_i is a parameter used to scale the effects of the repulsive potential, and $\rho\big(o_i(q)\big)$ is the shortest distance between o_i and any workspace obstacle. The workspace repulsive force for o_i is equal to the negative gradient of $U_{rep,i}$. Assuming our obstacles are all convex, and b is the point on the obstacle boundary that is closest to o_i, the function shall be given as

$$F_{rep,i}(q) = -\nabla U_{\mathrm{rep},i}(q) = \begin{cases} \eta_i \left(\dfrac{1}{\rho\big(o_i(q)\big)} - \dfrac{1}{\rho_0}\right) \dfrac{1}{\rho^2\big(o_i(q)\big)} \dfrac{o_i(q) - b}{\|o_i(q) - b\|} & ; \quad \rho\big(o_i(q)\big) \le \rho_0 \\ 0 & ; \quad \rho\big(o_i(q)\big) > \rho_0 \end{cases}$$

However, we would like to adjust the direction of the workspace repulsive force. Assume the obstacle is a rectangle and point is blocked by it on the way to the goal position, shown as Figure 1-1

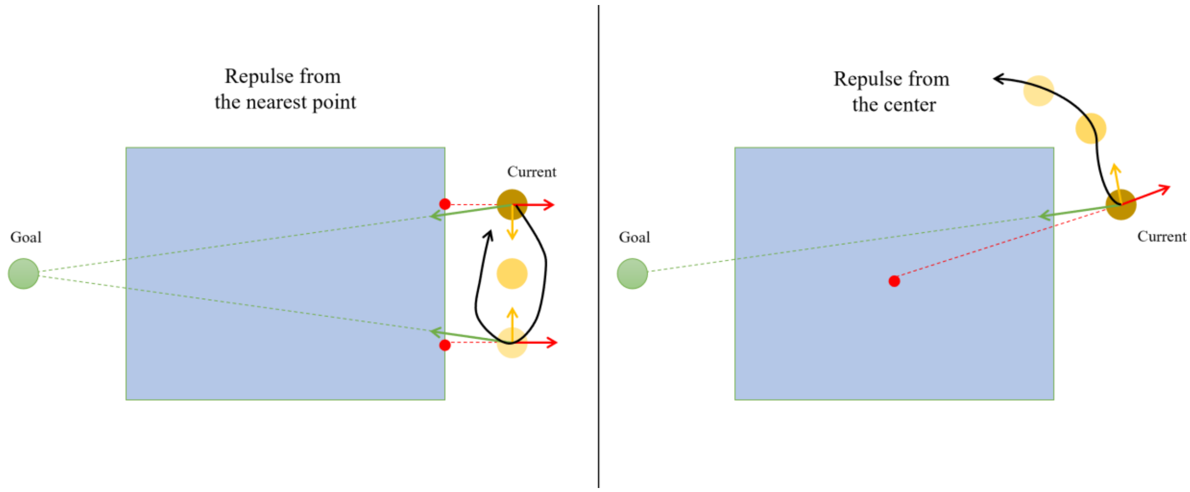

Figure 1-1 Repulse form the nearest point and from the center

In Figure 1-1, the green arrow is the attractive force, the red arrow is the repulsive force, and the yellow arrow is the total force. The left part shows that when the repulsive force is pointed from the nearest point of the obstacle, then we may easily fall into local minima. The right part shows that when the repulsive force is pointed from the center of the obstacles, it may generate a sideslip force to help the point cross over the obstacle. Therefore, setting c as the center of the obstacle, we define our function as

$$F_{rep,i}(q) = \begin{cases} \eta_i \left(\dfrac{1}{\rho\big(o_i(q)\big)} - \dfrac{1}{\rho_0} \right) \dfrac{1}{\rho^2\big(o_i(q)\big)} \dfrac{o_i(q) - c}{\|o_i(q) - c\|} & ; \quad \rho\big(o_i(q)\big) \le \rho_0 \\ 0 & ; \quad \rho\big(o_i(q)\big) > \rho_0 \end{cases}$$

1.3 Mapping forces to torques

By the linear identity of the gradient operation, we have

$$\begin{aligned} F_i(q) &= -\nabla U_i(q) = -\nabla\left(U_{\mathrm{att},i}(q) + U_{\mathrm{rep},i}(q)\right) \\ &= -\nabla U_{\mathrm{att},i}(q) - \nabla U_{\mathrm{rep},i}(q) = F_{\mathrm{att},i}(q) + F_{rep,i}(q) \end{aligned}$$

As we derived in the class (SHV p149), for o_i, if $\tau_i(q)$ denotes the vector of joint torques induced by the workspace force $F_i(q)$, then

$$\tau_i(q) = J^T_{v,o_i}(q) F_i(q)$$

in which $J_{v,o_i}(q) = [J_{v_1}(q) \cdots J_{v_i}(q)]$ is the linear velocity Jacobians for o_i, in which j^{th} column is

$$J_{v,o_{i_j}}(q) = z_{j-1} \times (o_i(q) - o_{j-1}(q))$$

Adding all the torques, we have

$$\tau(q) = \sum \tau_i(q)$$

1.4 Gradient descent planning

The symbolic representation of the robot is shown as Figure 1-2.

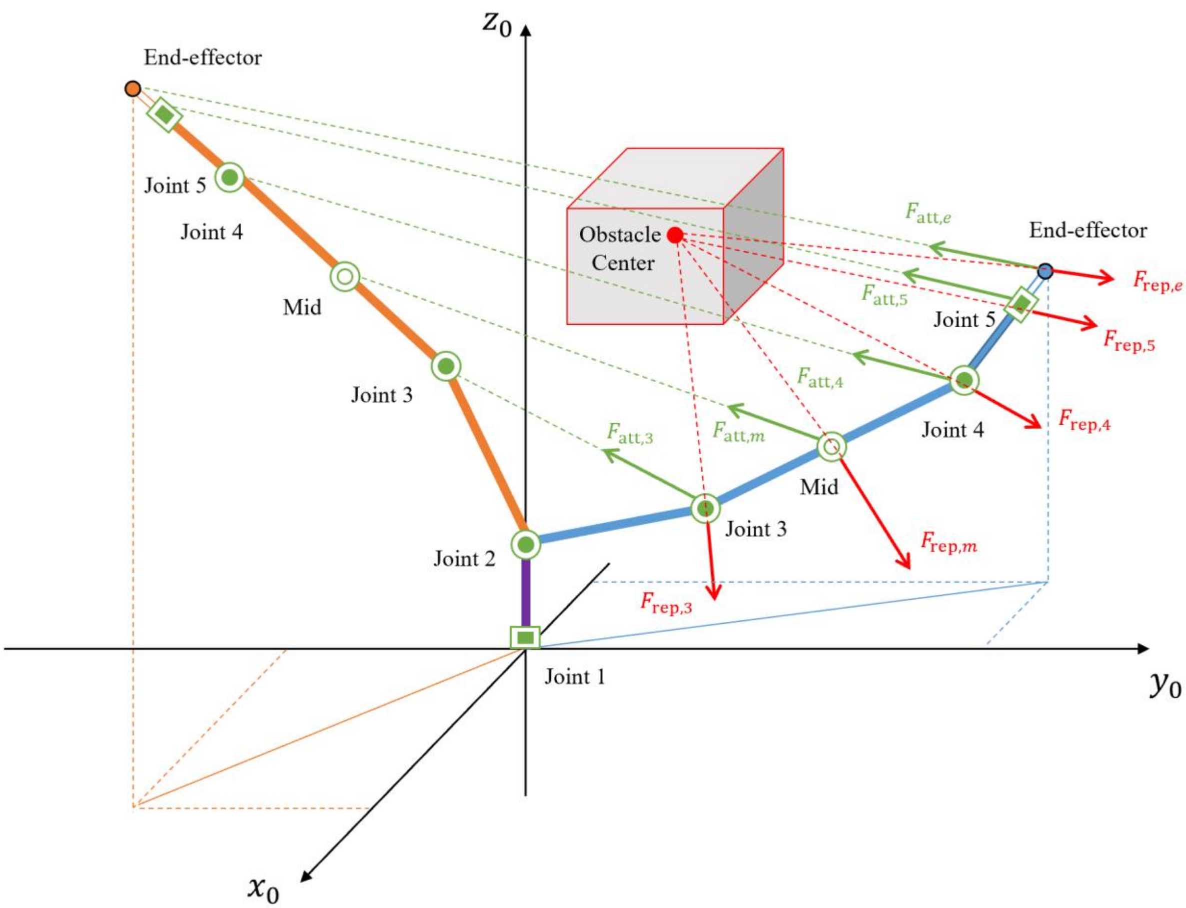

Figure 1-2 Symbolic representation of the robot

In Figure 1-2, the present configuration q is shown as the blue links and the final configuration q_f is shown as the orange links. The obstacle is shown as a red shaded box. The green arrows are the attractive forces, the red arrows are the repulsive forces.

In our planner, since joint 1 and joint 2 will not move through the trajectory, we manage to implement the field on joint 3 (o_4), joint 4 (o_5), joint 5 (o_6), end-effector (o_e), and the midpoint m of the link 3.

In terms of planning, we take gradient descent as the approach for solving optimization problems. The pseudocode is shown as followed.

Algorithm Gradient Descent

1. $q^0 \leftarrow q_s, i \leftarrow 0$
2. **IF** $\|q^i - q_f\| > \epsilon$

    $q^{i+1} \leftarrow q^i + \alpha^i \frac{\tau(q^i)}{\|\tau(q^i)\|}$

    $i \leftarrow i + 1$

    ELSE $return < q^0, q^1, \cdots q^i >$
3. **IF** stuck in a local minima

    $q' \leftarrow q^i + \text{random } q^\sigma$

    $q^{i+1} \leftarrow q'$

    $i \leftarrow i + 1$
4. **GO TO** 2

In which, α^i is a step parameter and since it is unlikely we will exactly reach the exact final configuration, we choose ϵ to be a sufficiently small constant based on the task requirement. q^σ is also a step parameter when it comes to a local minima.

As what is done in the Lab3, when the robot is moving, we are trying to avoid the obstacles in the workspace. The main difference between the simulated workspace and the physical workspace is that the arm may collide with the base support, so we treat the base support as an obstacle and add it to the map, described as $[x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}]$, which is $[-60, -60, -150, 60, 60, 10]$ after the measurement.

There is a MATLAB code *fieldPath.m* accomplishing the task listed in Appendix 1 and attached to the file.

2 Evaluation

2.1 Implement the planner in map_1

We set three tasks in map_1.

In the first task, we want to test the planner's performance when the path is relatively long, and the goal configuration have two joints (4 and 5) in the repulsive influence region.

We manage to move the robot from the middle above of the obstacles to the middle below of the obstacles, as in the configuration space, from $q_s = [0, -0.2\ , -0.5, 0, 0, 0]$ to $q_f = [0, 0.6, 0.4, -0.1, 0, 0]$. The outputted trajectory of our planner is shown as Figure 2-1. The light blue lines show the trajectory of the end positions.

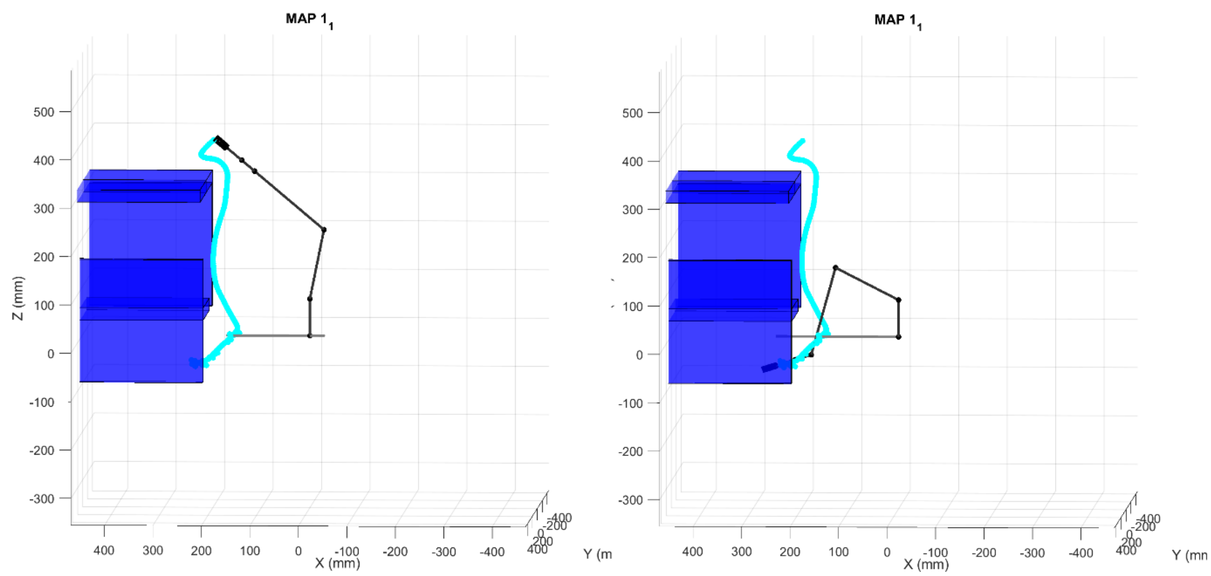

Figure 2-1 Trajectory of test 1.1

Measure the time, iteration number and local minima number, the result is shown as Table 2-1

Elapsed Time	Iteration Number	Number of Local Minima
$67.137\ s$	393	17

Table 2-1 Planner behavior on test 1.1

There is a video (Video 1) showing the simulated trajectory of test 1.1 and a video (Video 2) showing the physical trajectory of test 1.11 listed in Appendix 2.

In the second task, we want to test the planner's performance when the start position and final position is on the opposite side of the obstacle and is near to obstacle, and the goal configuration have one joint (4) in the repulsive influence region.

We manage to move the robot from the middle of the obstacles to the left of the obstacles, as in the configuration space, from $q_s = [0, 0\ , 0.3, 0, 0, 0]$ to $q_f = [0.7, 0.55, -0.3, -0.2, 0, 0]$. The outputted trajectory of our planner is shown as Figure 2-2.

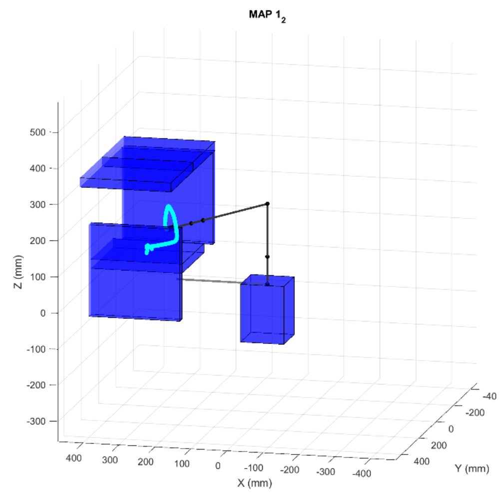

Figure 2-2 Trajectory of test 1.2

Measure the time, iteration number and local minima number, the result is shown as Table 2-2

Elapsed Time	Iteration Number	Number of Local Minima
$22.775\ s$	221	3

Table 2-2 Planner behavior on test 1.2

There is a video (Video 3) showing the simulated trajectory of test 1.2 listed in Appendix 2.

In the third task, we want to test the planner's performance when the end-effector runs into the overlapped influence area of two obstacles. and the goal configuration have joint 5 in the repulsive influence region.

We manage to move the robot from the middle of the obstacles to the below of the obstacles, as in the configuration space, from $q_s = [0, 0\ , 0, 0, 0, 0]$ to $q_f = [0, 0.6, 0.4, -0.1, 0, 0]$. The outputted trajectory of our planner is shown as Figure 2-3.

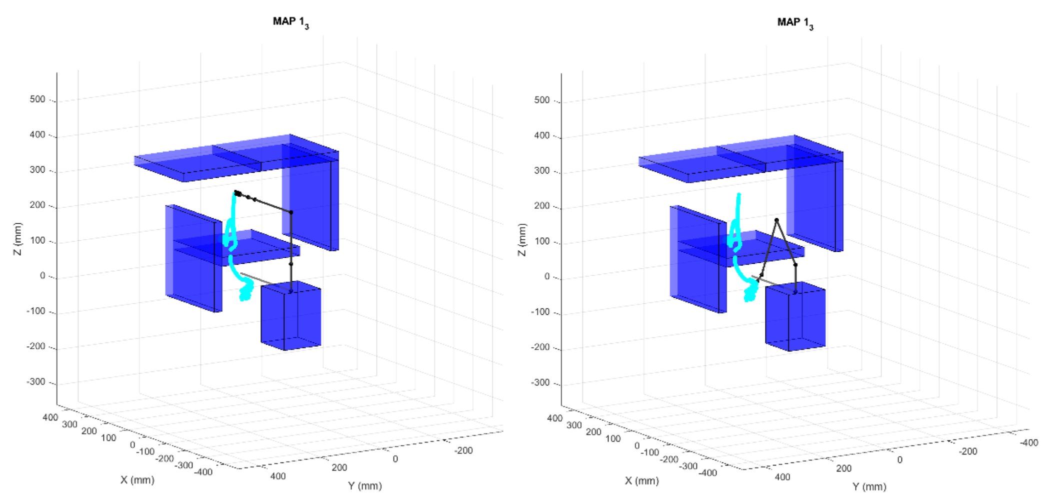

Figure 2-3 Trajectory of test 1.3

Measure the time, iteration number and local minima number, the result is shown as Table 2-3

Elapsed Time	Iteration Number	Number of Local Minima
$114.433\ s$	419	62

Table 2-3 Planner behavior on test 1.3

There is a video (Video 4) showing the simulated trajectory of test 1.3 listed in Appendix 2.

2.2 Implement the planner in map_2

In this task, we want to test the planner's performance when the start position and final position is on the opposite side of the obstacle and is near to obstacle, and the path may near the base, and the goal configuration have two joints (4 and 5) in the repulsive influence region.

We manage to move the robot from the middle of the obstacles to the below of the obstacles, as in the configuration space, from $q_s = [0, 0\ , 0, 0, 0, 0]$ to $q_f = [0, 0.2, 0.4,\ 0, 0, 0]$. The outputted trajectory of our planner is shown as Figure 2-4.

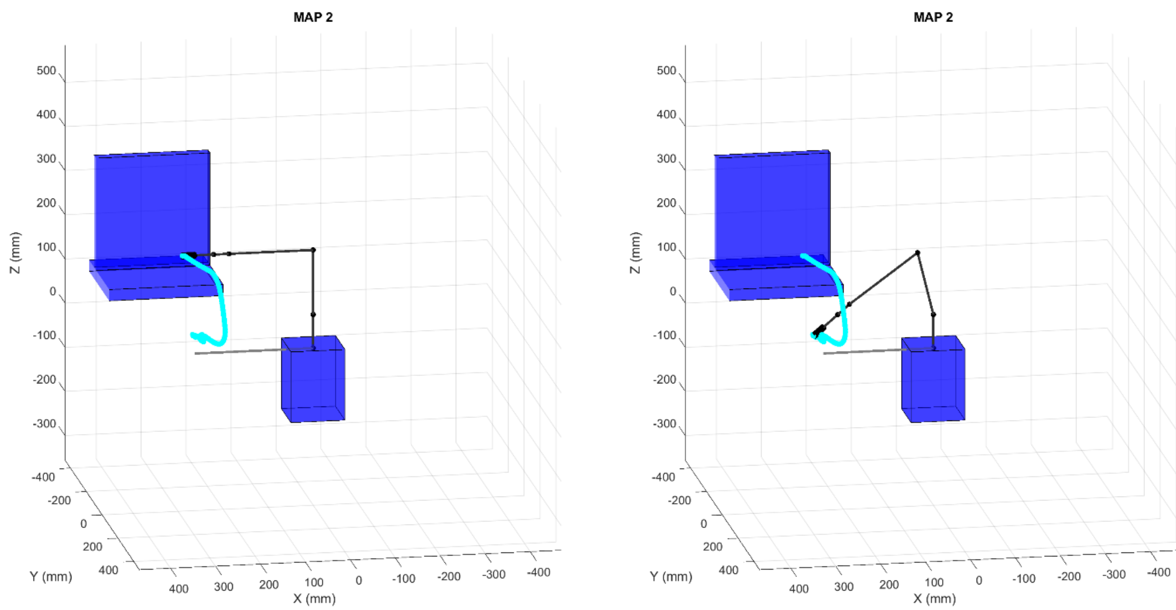

Figure 2-4 Trajectory of test 2

Measure the time, iteration number and local minima number, the result is shown as Table 2-4

Elapsed Time	Iteration Number	Number of Local Minima
$17.058\ s$	207	15

Table 2-4 Planner behavior on test 2

There is a video (Video 5) showing the simulated trajectory of test 2 listed in Appendix 2.

2.3 Implement the planner in map_3

In this task, we want to test the planner's performance when the trajectory is relatively long and encounters some narrow paths, and the goal configuration have no joints in the repulsive influence region.

We manage to move the robot from the middle of the obstacles to the below of the obstacles, as in the configuration space, from $q_s = [0, 0\ , 0, 0, 0, 0]$ to $q_f = [1.2, 0.3, \text{-}0.6, -0.3, 0, 0]$. The outputted trajectory of our planner is shown as Figure 2-5.

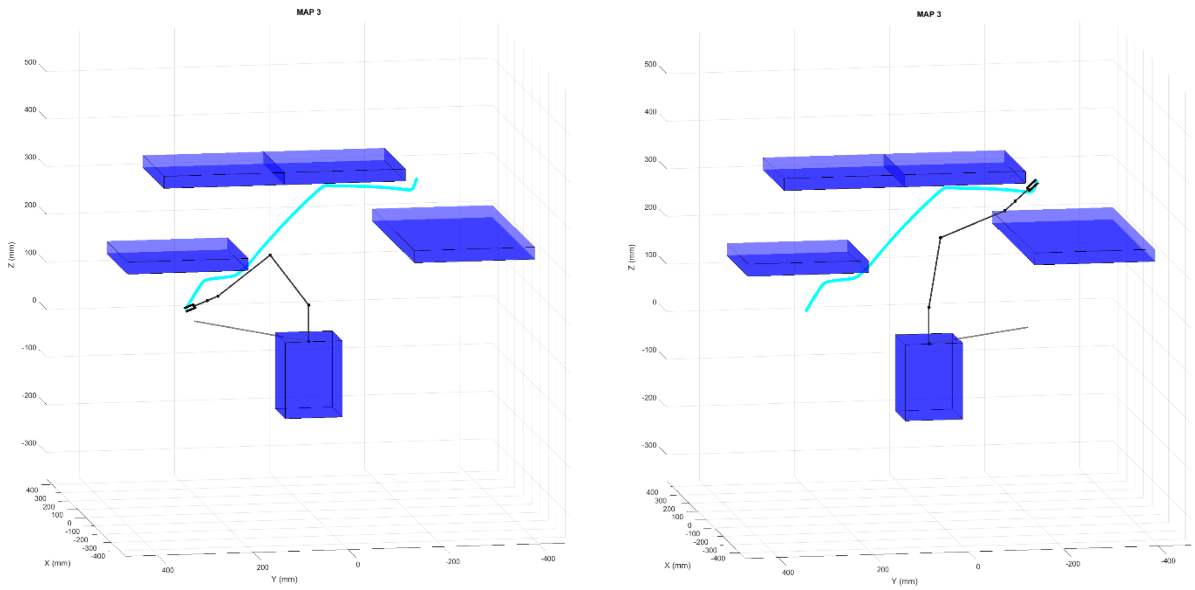

Figure 2-5 Trajectory of test 3

Measure the time, iteration number and local minima number, the result is shown as Table 2-5

Elapsed Time	Iteration Number	Number of Local Minima
23.767 s	183	0

Table 2-5 Planner behavior on test 3

There is a video (Video 6) showing the simulated trajectory of test 2 listed in Appendix 2.

2.4 Implement the planner in map_4

In this task, we want to test the planner's performance when the start and end configuration is near singular and encounters some narrow paths, and the goal configuration have two joints (4 and 5) in the repulsive influence region.

We manage to move the robot from the middle of the obstacles to the below of the obstacles, as in the configuration space, from $q_s = [0, 0\ , -1.6, 0, 0, 0]$ to $q_f = [1.1, 0.1, \text{-}0.5, 0.5, 0, 0]$. The outputted trajectory of our planner is shown as Figure 2-6.

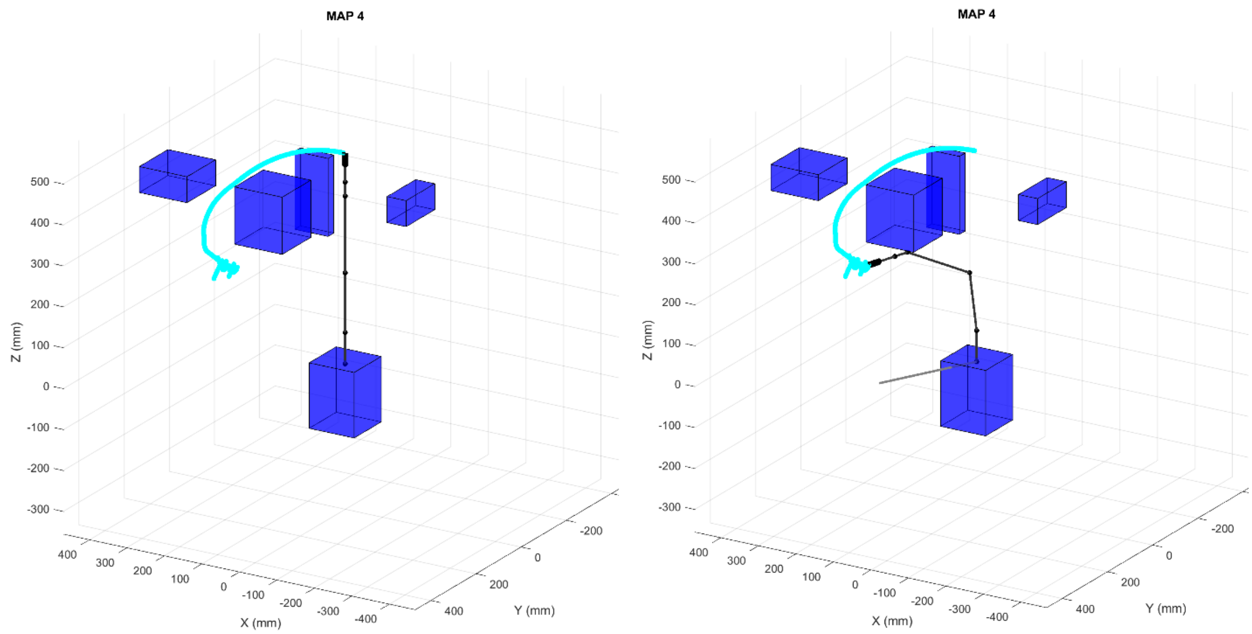

Figure 2-6 Trajectory of test 4

Measure the time, iteration number and local minima number, the result is shown as Table 2-6

Elapsed Time	Iteration Number	Number of Local Minima
$40.203\ s$	267	25

Table 2-6 Planner behavior on test 4

There is a video (Video 7) showing the simulated trajectory of test 2 listed in Appendix 2.

3 Analysis

3.1 Strategy for choosing parameters

There are several parameters in the planners.

(1) Attractive parabolic distance parameters d
d is the distance defines the transition from conic well potential to parabolic well potential field. The attractive force become smaller if the current point lies in parabolic influence area measured by the distance d. The main intention of this transition is given by the following explanation. Consider the plane formed by link 2 and link 3, which is shown as Figure 3-1.

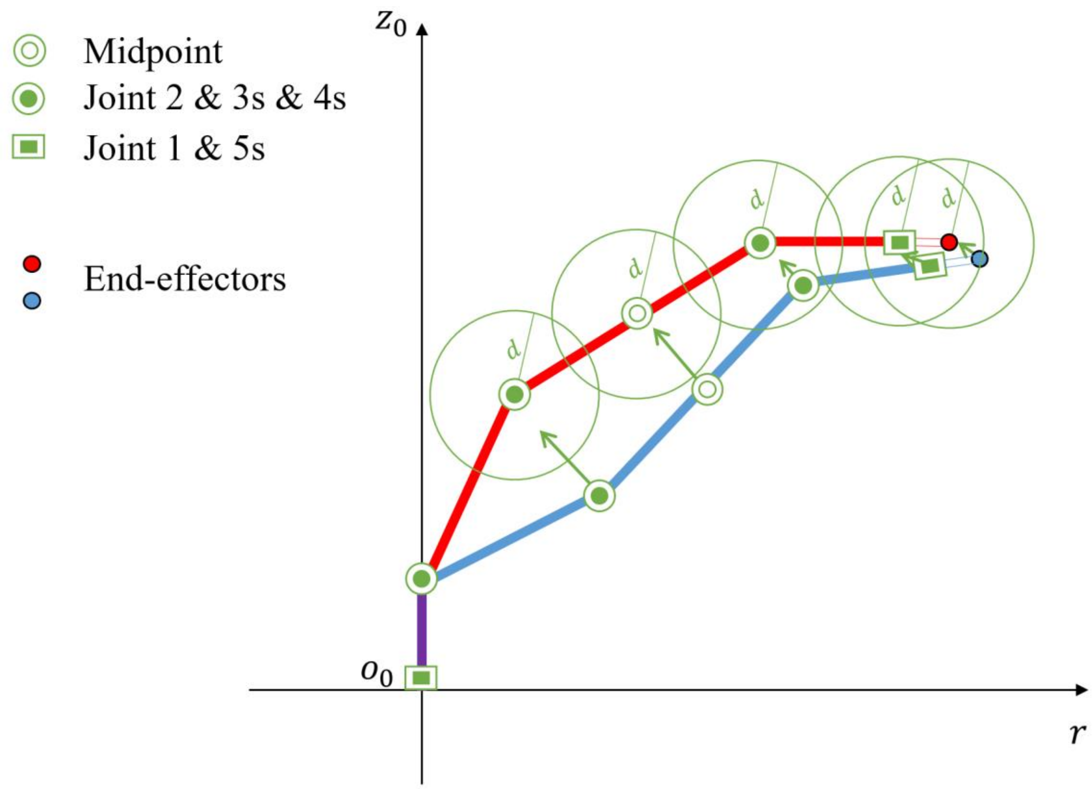

Figure 3-1 Projecting onto the plane formed by link 2 and link 3

In Figure 3-1, the red links is the goal configuration, the blue links is the current configuration, and the green arrow are the attractive forces.. In such configuration, joint 4, joint 5 and end-effector are near their goal positions while joint 3 and midpoint m are still relatively far away from their goal. Joint 4, joint 5 and end-effector lie in the parabolic influence area, thus are exerted forces by the parabolic potential, while the joint 3 and midpoint m are exerted forces by the conic potential. Therefore, the attractive forces on joint 3 and midpoint m would be greater than the other attractive forces, thus “squeeze” the arm to its goal configuration.

We interpret that when a joint lies in parabolic influence area, such joint is near its goal position. Therefore, we set d to be relatively small.

(2) Repulsive influence distance parameters ρ_0
ρ_0 is be the distance of influence of an obstacle. We do not want any obstacles' region of influence to include any goal positions, which can be satisfied by choosing the suitable goal configurations. Also, we try to avoid the overlapping regions of influence for different obstacles as much as we can. The ρ_0 we set for 4 maps are shown as Figure 3-2, in which the red line sketch are the repulsive regions of influence.

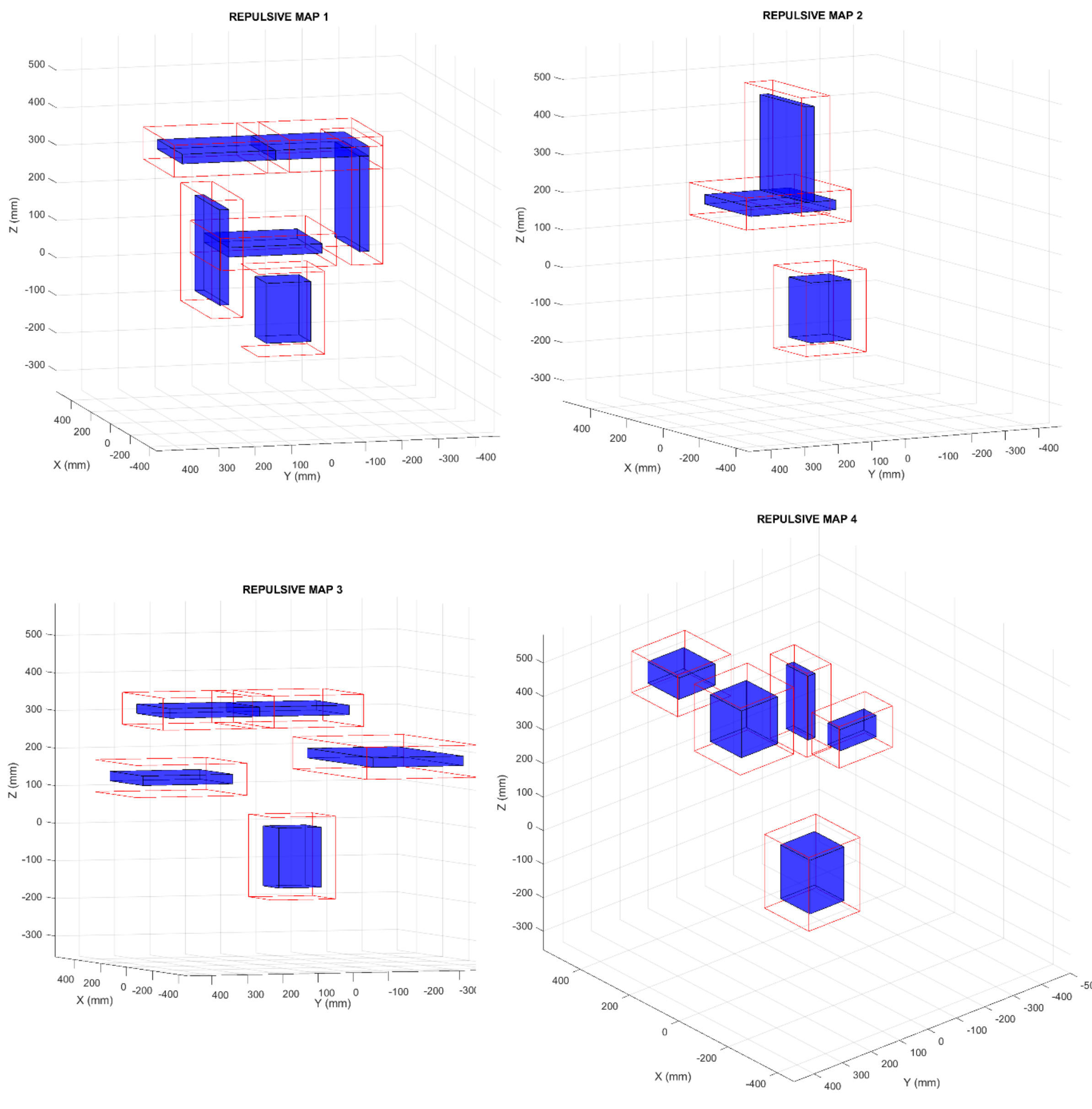

Figure 3-2 Obstacles' regions of influence of 4 maps

However, there are some situation where the overlapping can not be avoided just by adjusting the magnitude of ρ_0, which are the intersection area of two obstacles, shown in map_1, map_2, and map_3.

(3) Attractive forces parameters ζ_i and repulsive forces parameters η_i

ζ_i is controls the relative influence of the attractive potential for control point o_i. We assign a larger weight to one of the o_i than to the others, producing g a "follow the leader" type of motion. In our cases, we set ζ_e to be larger so that the end-effector could be the leader. Because our attractive forces are unit outside of the small region of parabolic potentials, and in most cases, the end-effector's distance to its goal position is longer than other sampled points.

η_i controls the relative influence of the repulsive potential for o_i. We assign a much smaller weight to one of the o_i which its goal is near the obstacles, to avoid having these obstacles repel the robot from the goal.

The specific values for ζ_i and η_i is relative, since the torque will be normalized by the planner. We just need to make sure that the values of η_i are strong enough to repel the attractive forces when the sampled points are near the obstacles.

(4) Iteration step parameters α^i and random walk step q^σ

α^i is a step parameter for each iteration. In our cases, we set α^i to be a very small constant so that the robot is not allowed to "jump into" obstacles.

q^σ is the step parameter when it comes to a local minima. In one local minima, we random q^σ under the step which is a little bigger than α^i, and will enlarge its value if it does not works.

(5) Error parameters ϵ and local minima condition parameter ϵ^σ

Because it is unlikely that we will ever exactly satisfy the final configuration, we choose ϵ to be sufficiently smaller to make it look like the same.

We choose ϵ^σ to be a little bit larger the α^i, since when the actual step is smaller α^i, it means that the torque vector lose rank, therefore may stuck in a local minima.

The parameter setting for the six tasks above is given as Table 3-1 and listed in the MATLAB codes.

Task	1.1	1.2	1.3	2	3	4
d	10	10	10	10	10	10
ρ_0	30	30	100	30	30	30
$\zeta_{3/m/4/5}$	0.001	0.001	0.001	0.001	0.001	0.001
ζ_e	0.1	0.1	0.1	0.1	0.1	0.1
$\eta_{3/m}$	4000	4000	4000	4000	4000	4000
η_4	40	40	4000	40	40	40
η_5	40	4000	40	40	4000	40
η_e	4000	4000	4000	4000	4000	4000
α^i	0.01	0.01	0.01	0.01	0.01	0.01
q^σ	0.02	0.02	0.02	0.02	0.02	0.02
ϵ	0.1	0.1	0.1	0.1	0.1	0.1
ϵ^σ	0.012	0.012	0.012	0.012	0.012	0.012

Table 3-1 Parameters for each tasks

d is the same for each task.

ρ_0 is the same except for task 1.3, where we want to make the robot away from the left obstacles/

ζ_i is the same for each tasks we set ζ_e to be the biggest one.

The smallest η_i for each task is set for closest to obstacles final joint in each task. Some tasks have two smallest η_i because that respective two final joints are both close to the obstacles.

α^i, q^σ, ϵ, and ϵ^σ is the same for each task.

3.2 The good-at tasks and bad-at tasks

The performance of the planner in each task is summarized as Table 3-2.

Task	Final Joints in Repulsive Region	Elapsed Time	Iteration Number	Number of Local Minima
1.1	$4/5$	$67.137\ s$	393	17
1.2	4	$22.775\ s$	221	3
1.3	5	$114.433\ s$	419	62
2	$4/5$	$17.058\ s$	207	15
3	null	$23.767\ s$	184	0
4	$4/5$	$40.203\ s$	267	25

Table 3-2 Performance of the planner in each tasks

From Table 3-2, we can see that task 1.2, task 2 and task 3 took relatively shorter time, have little iteration number and little local minima. The planner works well when the number of joints in the repulsive influence region in the final configuration is relatively limited to 1 or 0.

We can also see that task 1.3 took the most time, iteration number and encounter most local minima. It is because that the trajectory went through the area in which the repulsive regions of different obstacles are overlapped, it is the situation where our planner works bad at.

3.3 Compare to RRT trajectory planning

In our Lab3, we use RRT trajectory planning for the similar task 3. The compared result is shown as Table 3-3 and Figure 3-3.

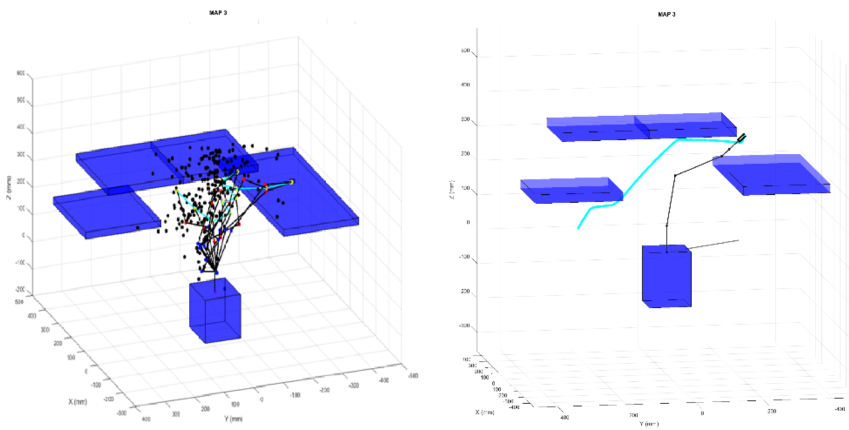

Figure 3-3 Comparation of RRT and potential field in task 2

Planner	Average Time	Successful Rate
Potential Field	17.058 s	100%
RRT	49.332 s	69%

Table 3-3 Comparation of RRT and potential field in task 2

When the trajectory is long and involves some narrow paths, the potential field works better than RRT.

We use RRT for another similar task 1.3. The compared result is shown as Table 3-4 and Figure 3-4.

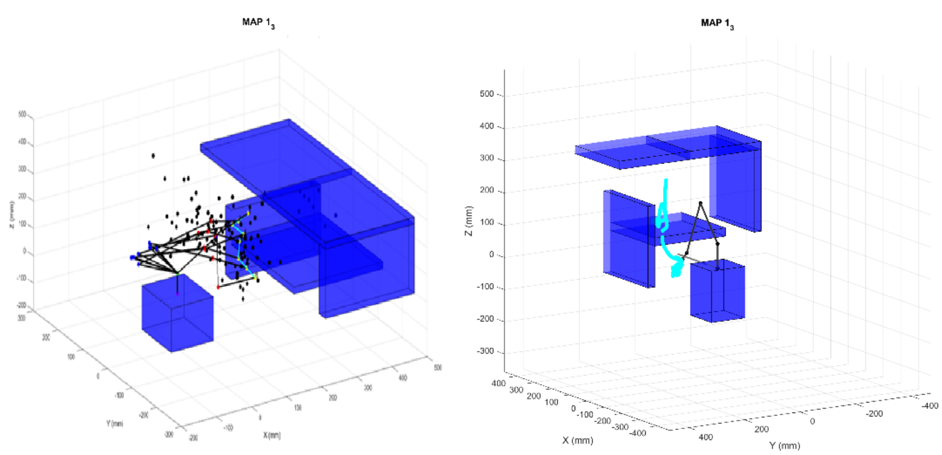

Figure 3-4 Comparation of RRT and potential field in task 1.3

Planner	Average Time	Successful Rate
RRT	$2.170\ s$	100%
Potential Field	$114.433\ s$	$< 100\%$

Table 3-4 Comparation of RRT and potential field in task 1.3

the trajectory went through the area in which the repulsive regions of different obstacles are overlapped, and when the end-configuration is near the obstacles, RRT works better than potential field.

3.4 Future work

We want to do more about parameter setting.

In this lab, ρ_0 is the same for different obstacles, we can adjust them to be different for different tasks. ζ_i and η_i is constant through the trajectory, we can make it as a function of iteration.

We would also to test more on the physical robot, to use Jacobian to compensate for the gravity error.

Appendix 1 Code Overview

1. *calculateFK_sol.m*	predefined compute forward kinematics
2. *calJacobian.m*	compute Jacobian
3. *calJacobian5.m*	compute Jacobian for joint 5
4. *calJacobianm.m*	compute Jacobian for midpoint
5. *loadmap.p*	load map
6. *plotnap.p*	plot map
7. *lynxInitializeHardware.m*	predefined lynx function
8. *lynxServo.m*	predefined lynx function
9. *lynxServoSim.m*	predefined lynx function
10. *lynxStart.m*	modified lynx function
11. *lynxVelocityPhysical.m*	predefined lynx function
12. *lynxServoPhysical.m*	predefined lynx function
13. *lynxVelocitySim.m*	simulate velocity control
14. *lynxVelocitySimO.m*	simulate velocity control with orientation control
15. *manipulability.m*	show the manipulability in different configurations
16. *plotLynx.m*	plot simulated configuration in 3D space (modified)
17. *potentialField*	planner

Appendix 1 Video Overview

Video 1 https://youtu.be/C4DzhLzQ2bQ	task 1.1
Video 2 https://youtu.be/UmkmGf4V2I0	task 1.1 physical
Video 3 https://youtu.be/sa2WNcnFUPk	task 1.2
Video 4 https://youtu.be/-CBwCv1O9lM	task 1.3
Video 5 https://youtu.be/lEIO7B_LjEs	task 2
Video 6 https://youtu.be/Nn4sLk7KDJU	task3
Video 7 https://youtu.be/ngPg_lJL_9o	task 4