Robot Motion Planning
Spring 2024
Final Project Report

Delphine Tan, Mariana Smith, Sameer Khan

Due May 10th, 2024

Task 1: Planar Serial Manipulator

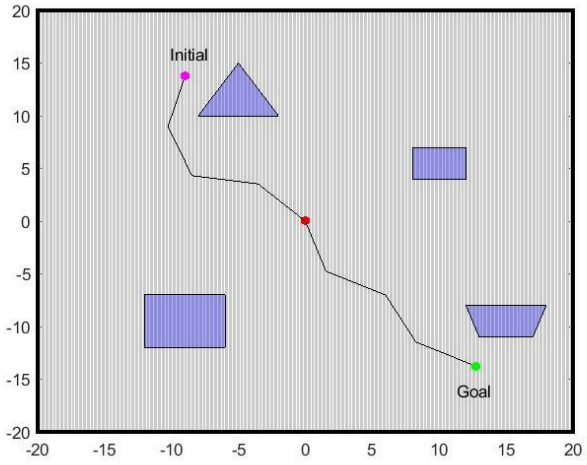

Figure 1: Plotted workspace, obstacles, and initial and goal configurations for Task 1.

Robot dimensions: The robot used for this task consists of n=4 links, where each link has a length of 5.

Workspace dimensions: The 2D workspace is defined as a square with a width of 2*xmax and a length of 2*ymax. The variables "xmax = 20" and "ymax = 20" are included as user-defined inputs to the planning functions.

Obstacle dimensions: Four convex polygonal obstacles were created in the 2D plane, and saved within a cell array "B". Each obstacle is stored in one element of B, where the element is a 2xm matrix (where m is the number of vertices in the obstacle).

Initial and goal configurations: The initial and goal configurations "qi" and "qg" are column vectors consisting of n=4 joint angles. The variables "qi = [3*pi/4; pi/5; -pi/3; -pi/5]" and "qg = [-2*pi/5; pi/4; -pi/5; pi/5]" are included as user-defined inputs to the planning functions.

<u>Task 1a: PRM</u>

Other inputs to the PRM function are:

- `num_vertices = 150`, the scalar number of random vertices to generate before planning.
- `K = 6`, the scalar number of closest neighbors to connect with edges.

The outputs of the PRM function are:

- `V_prm`, a cell array of the generated graph vertices.
- `E_prm`, a cell array of the generated graph edges.
- `path_prm`, a 4xn matrix of the shortest path (generated with Dijkstra algorithm).

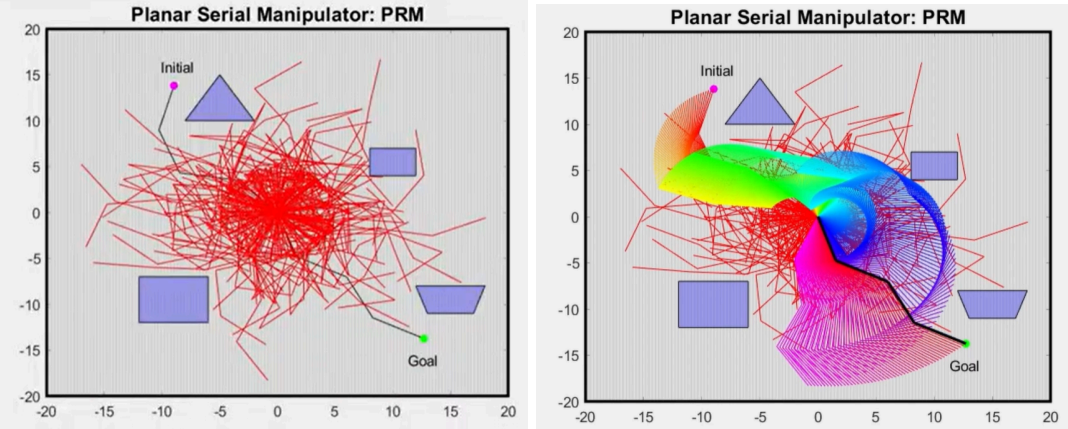

Figure 2: A) generated PRM configurations, and B) selected robot path.

Task 1b: RRT

Other inputs to the RRT function are:

- `dq = 0.5`, the chosen step size away from some q.
- `near_qg_thresh = 2`, the threshold value to determine "near-ness" to qg.

The outputs of the RRT function are:

- `V_rrt`, a cell array of the generated graph vertices.
- `E_rrt`, a cell array of the generated graph edges.
- `path_rrt`, a 4xn matrix of the shortest path (generated with Dijkstra algorithm).

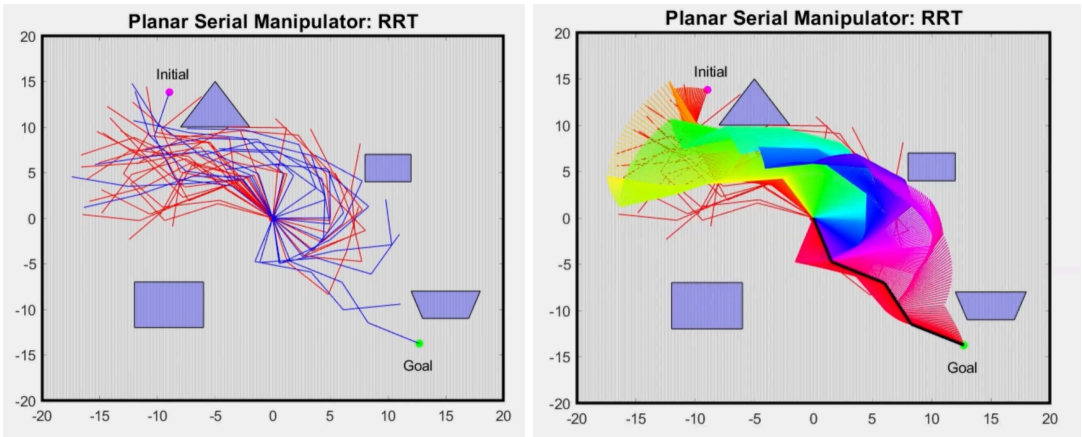

Figure 3: A) generated RRT configurations, and B) selected robot path.

Task 2: Planar Rigid Robot

Task 2 focuses on motion planning for the planar holonomic rigid robot. The robot in this task is a rectangular rod/stick. The potential field method is applied to plan from the initial to the goal position and to avoid obstacles.

Obstacle and World Dimensions:
The robot has four control points and its configuration space is SE(2). The dimension of the robot is 10 x 2. The control points are located at the corners of the stick. The body frame of the robot is attached to the center of mass

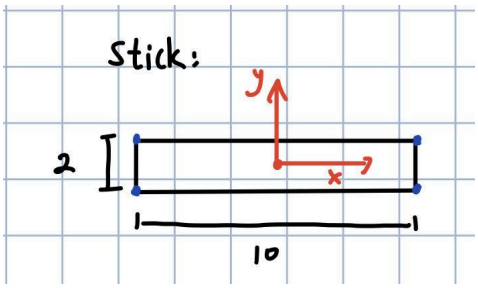

Figure 4: Task 2 robot

For the world, there are three obstacle: two triangles and one rectangle. The robot has to avoid all of them when traveling from the initial configuration to the final configuration. The dimension of the world is 50 x 50. The potential field acts as force/torque, rotating and translating the robot. Each control point then gets a force and torque from the potential field and adds up to find the total force and torque acting on the robot.

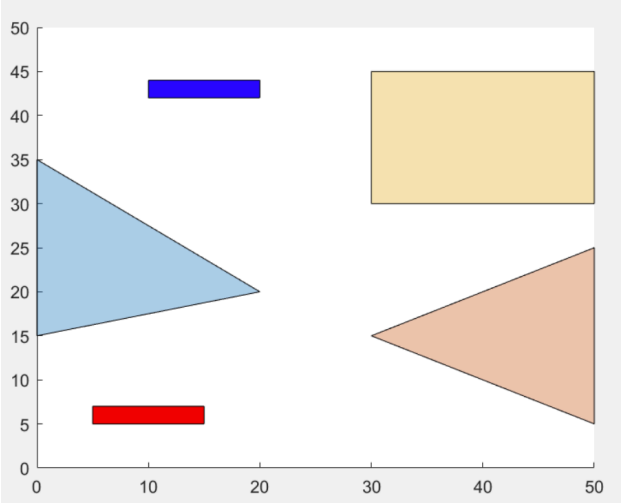

Figure 5: Task 2 World Configuration
(red = robot's initial position, blue = robot's final position)

Parameters:
The potential method used in this task has the following parameters:

α = 0. 12 (gradient descent step)
max steps = 2000 (maximum # of steps for gradient descent)
η = 500 (repulsive potential force coefficient)
ζ = 0. 06 (attractive potential force coefficient)
σ = 10 (maximum distance for the existence of repulsive force from obstacles)
d (step for torque)

Results:
The total steps for this planning is 345 steps. The robot was able to avoid all the obstacles and reached the goal position.

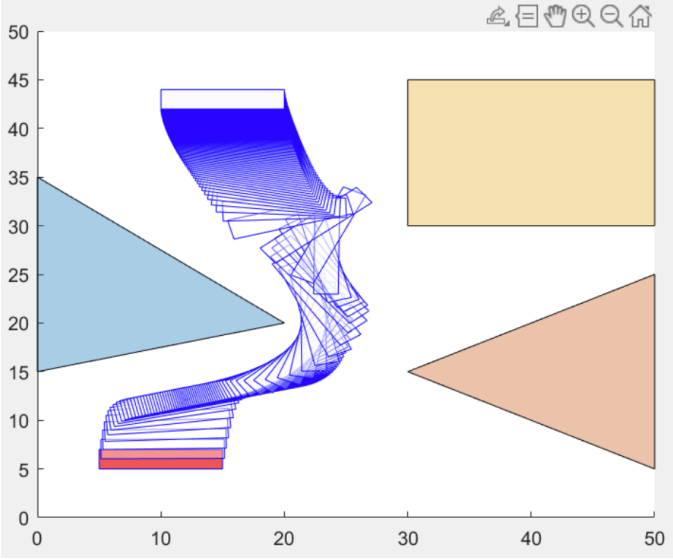

Figure 6: Task 2 Result

Task 3: Simple Model for Flexible Needle

The goal of Task 3 is to perform path planning for an asymmetric, flexible needle using a nonholonomic unicycle model in a planar scenario. RRT is used to construct a search tree using these constraints and then the path is found using Dijkstra's algorithm.

Model Derivation:

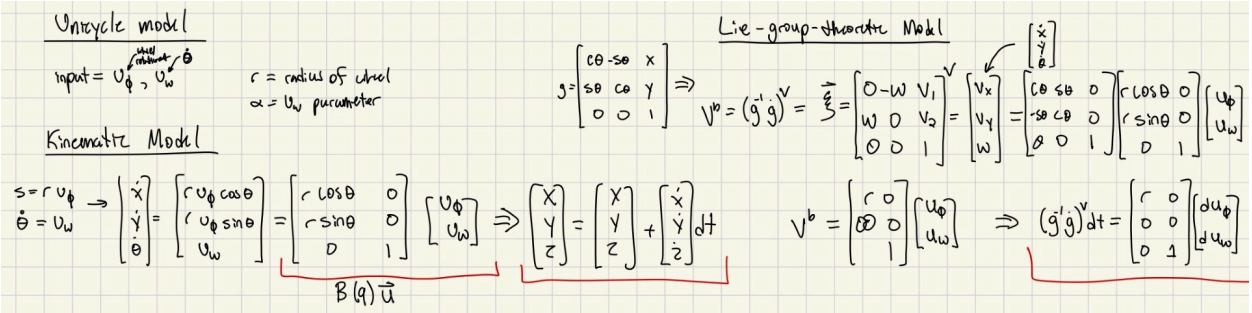

Figure 7: Derivation for unicycle model

Obstacle and World Dimensions:

The robot is modeled in the figures as an isosceles triangle with base length 2 and height 4. The center and body frame of the needle tip is placed at the center of the base of the triangle and the tip is the point (4,0) away from the center.

The world is a 100x100 square with the bottom left corner containing the world frame at (0,0). There are three obstacles in the world. The first is a 30x30 square in the middle of the space. There is also a large triangle at the top of the space and a smaller rectangle on the right edge of the world space. These dimensions are described in "bounds" for the world bounds and "B" for the obstacles.

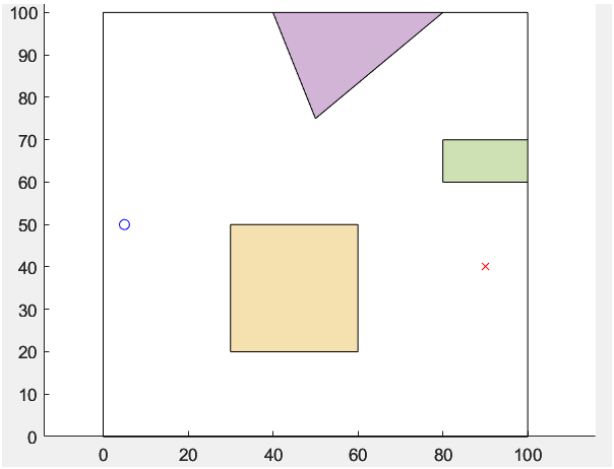

Figure 8: Task 3 World Configuration with qI(blue circle) and qG(red x)

Parameters:

The flexible needle model has the following parameters:

Inputs:

- alpha = .3: robot rotation rate parameter
- qI = [5;50;0]: initial state
- qG = [90;40;0]: goal state
- n =2000: max number of iterations for RRT
- dt = .5: time step
- r = 2: wheel radius
- tip = [4;0]: body frame location of tip of needle
- B: obstacle cell array
- x_max = 100: max x bound
- y_max = 100: max y bound

Outputs:

- path: path of states from qI to Xg
- V: all vertices of RRT
- E: edges of RRT in parent/child form with inputs to each state

Other Parameters:

- u_phi = 1: rotation rate of the wheels
- u_w = [0,alpha,-alpha]: set of inputs for robot rotation rate
- Xg = 3: Goal set for qG; a radius of Xg around qG is part of the goal set

Results:

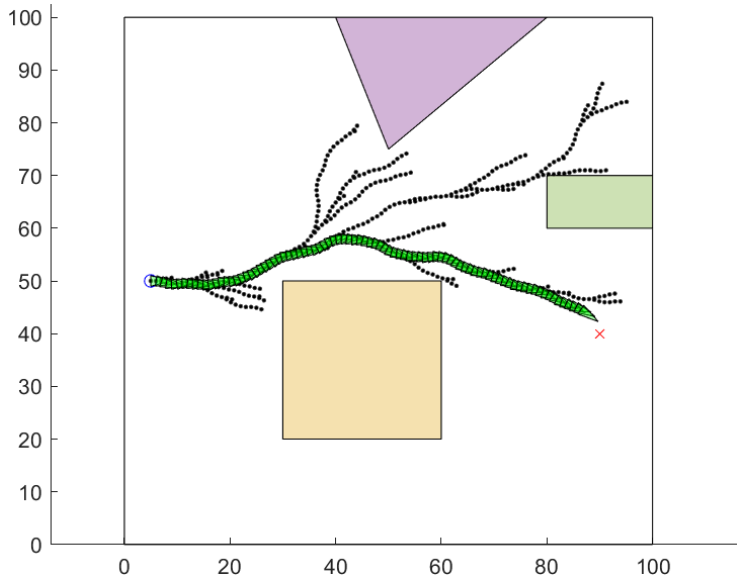

Figure 9: Task 3 possible result
(black dots = found states from RRT, green triangles = path)

Task 4 (Extra Credit): RRT for a UR5 Robot

We expand upon the work in Task 1 to conduct a RRT plan for a 3-dimensional UR5 robot.

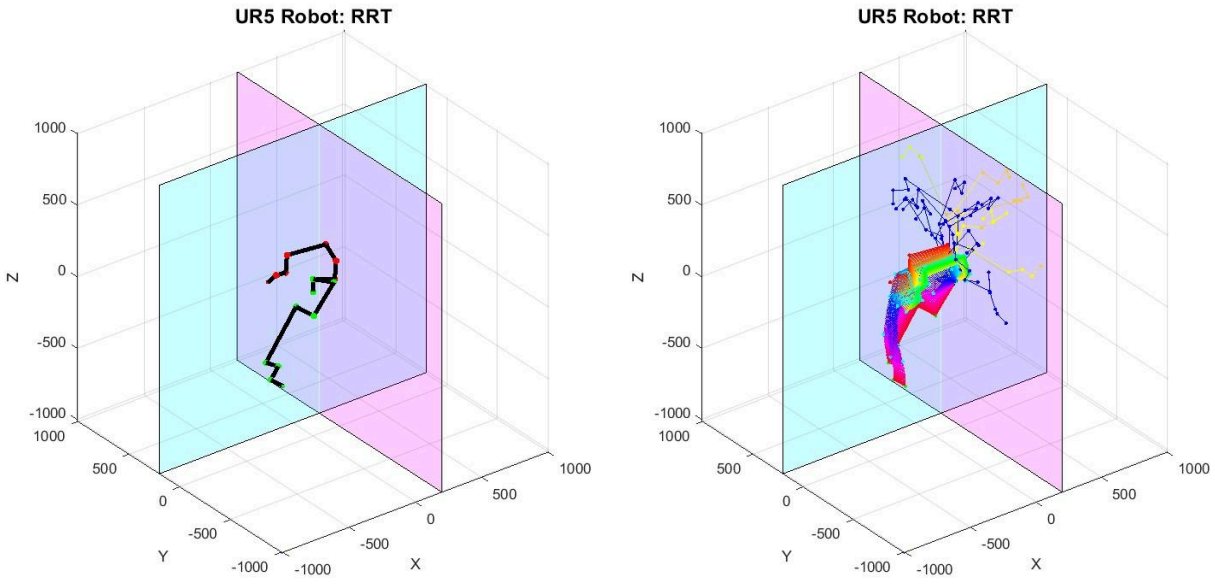

Figure 10: A) Initial (red) and goal (green) configurations, with plane obstacles. B) New configurations with collision (yellow), new configurations without collision (blue), and robot path (rainbow).

Problem setup:

- The robot is the UR5 robot with link lengths [89; 130; 425; 120; 392; 110; 95; 83] mm.
- The workspace is defined as a cube with xmax = ymax = zmax = 1010 mm.
- Each configuration is expressed as a vector of 7 joint angles; qi and qg are both derived by adding angular offsets to the q_home condition. The vertices used for plotting are calculated in the plot_config() and get_vertices() functions using homogeneous transformations.
- The obstacles are planar. One obstacle is the plane y=200, and the second obstacle is the plane x=200. Collisions are detected by checking whether each joint falls on the far side of the obstacle plane.

The remainder of the RRT code follows closely from the Task 1 workflow. **Note: I have not used code from another class for Task 4, with the exception of my simple ROTX, ROTY, and ROTZ functions which trivially generate rotation matrices (from RDKDC). All other code for Task 4 was written specifically for this project.**

Workload Distribution

Mariana Smith: Task 1 code and Task 1 report section, Task 4 and Task 4 report section
Delphine Tan: Task 2 and Task 2 report section
Sameer Khan: Task 3 and Task 3 report section