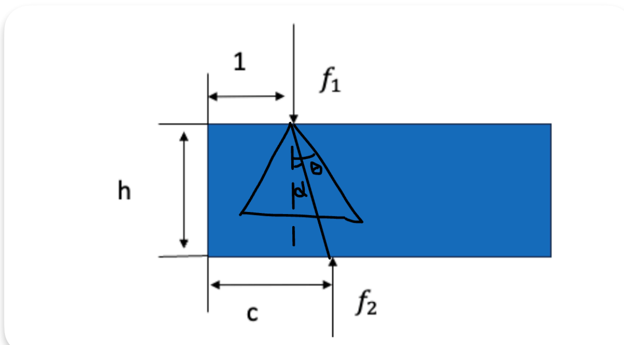
CS5478 Homework 2

Jinbin Bai
jinbin.bai@u.nus.edu

October 15, 2023

1 1.



As shown in the figure above. α is the angle between the line connecting the points of f_1, f_2 and the vertical direction. θ is the angle of friction cone.

To achieve force closure, α should be less than θ.

We know that $\tan\theta = \mu$ and $\tan\alpha = \frac{|c-1|}{h}$

Then $\frac{|c-1|}{h} < \mu$

So $\mu > \frac{|c-1|}{h}$

2 2.

First, the friction force of f_1 should be larger than f_3. So $\mu F \geq F$, then $\mu \geq 1$.

Then, the torque equation for force closure is:

$$\tau_1 - \tau_2 \geq 0$$

$$d_3 \times F - d_2 \times F \geq 0$$

(where d_i is the vector of f_i to CoM.)

Given c and h:

$$\mu F - 0.25F \geq 0$$

$$\mu \geq 0.25$$

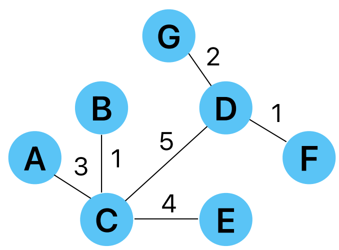
So μ can be 1.

3 3.

3.1 a.

	A	B	C	D	E	F	G
V_0	0	5	3	∞	∞	∞	∞
V_1	0	4	3	8	7	∞	17
V_2	0	4	3	8	7	9	10
V^*	0	4	3	8	7	9	10

3.2 b.



4 4.

4.1 a.

The dimension of the configuration space for this system is 2. Each joint corresponds to a degree of freedom, and there are two revolute joints, allowing the robot's end-effector to have two-dimensional motion within the plane.

4.2 b.

The dimension of the configuration space for this system is 6. Each mobile robot has two degrees of freedom for translation (x and y coordinates) and one degree of freedom for rotation (orientation). Therefore, for two mobile robots, there are a total of 2 * 3 = 6 degrees of freedom in the configuration space.

4.3 c.

The dimension of the configuration space for this system is 18. For each manipulator, it has 6 revolute joints, so the configuration space for one manipulator is 6-dimensional. Since there are two manipulators attached to the UAV, the total dimension of the configuration space is 2 * 6 + 6 (UAV) = 18.

5 5.

5.1 a.

Parameterization using angles: We can use two angles, θ and ϕ to represent the direction of the line, and a point (x_0, y_0, z_0) on the line. So, the configuration space C can be parameterized as $(\theta, \phi, x_0, y_0, z_0)$.

Parameterization without angles: We can use a unit vector $u = (u_x, u_y, u_z)$ to represent the direction of the line, and a point (x_0, y_0, z_0) on the line.

5.2 b.

For parameterization using angles, it is 5. For parameterization without angles, it is 6.

5.3 c.

For straight-line segment s, parameterization using angles is 6 and parameterization without angles is 7. Specifically, we need an additional parameter to specify its length. And the modified parameterizations would be $(\theta, \phi, x_0, y_0, z_0, l)$ and $(u_x, u_y, u_z, x_0, y_0, z_0, l)$, respecitively.

6 6.

6.1 a.

If the algorithm calls LINK for every pair of roadmap nodes, we have n milestones, and each milestone must be linked to every other milestone, resulting in a total of $n * (n-1)$ LINK calls. The asymptotic upper bound on the number of calls to LINK in this case is $O(n^2)$.

6.2 b.

Because the milestones are distributed roughly uniformly in C, we can assume there is πn^2 area to call LINK, and there will be $\pi n^2 t^2$ milestones, resulting in a total of πn LINK calls. So the asymptotic upper bound on the number of calls to LINK in this case is $O(n)$.

7 7.

7.1 a.

3 dimensions. The hybrid A* algorithm associates with each node a continuous configuration $q = (x, y, \theta)$.

7.2 b.

f-value = cost-to-come + heuristic estimate, so

Node A: $f(A) = 3.7 + 3.2 = 6.9$
Node B: $f(B) = 2 + 4.7 = 6.7$
Node C: $f(C) = 2.5 + 4 = 6.5$
Node D: $f(D) = 4 + 2 = 6$
So, the priority queue contains the following nodes and their associated f-values:
D with f-value 6
C with f-value 6.5
B with f-value 6.7
A with f-value 6.9

8 8.

8.1 a.

To show that the heuristic function $h(x) = \max\{h_1(x), h_2(x)\}$ is admissible, we need to prove that it satisfies two properties:

1. $h(x) \geq 0$ for all states x: This is generally true because both $h_1(x)$ and $h_2(x)$ are admissible, which means they provide non-negative estimates of the cost to reach the goal.

2. $h(x) \leq h^*(x)$ for all states x, where $h^*(x)$ is the true, optimal cost to reach the goal from state x: We can prove this property for $h(x)$ by considering the following inequalities:

First, since $h_1(x)$ is an admissible heuristic, we have $h_1(x) \leq h^*(x)$ for all states x. Second, since $h_2(x)$ is an admissible heuristic, we have $h_2(x) \leq h^*(x)$ for all states x.

Now, when we take the maximum of two non-negative values (in this case, $h_1(x)$ and $h_2(x)$), the result can only be greater than or equal to both values. Therefore, $h(x) = \max\{h_1(x), h_2(x)\}$ satisfies the inequality $h(x) \leq h^*(x)$ for all states x. This proves that $h(x)$ is admissible.

8.2 b.

I would use $h(x)$.

1. Admissibility: We've shown that $h(x) = \max\{h_1(x), h_2(x)\}$ is admissible.

2. Optimality: $h(x)$ tends to be a more informed and conservative estimate compared to $h_1(x)$ and $h_2(x)$ individually. By taking the maximum of the two heuristics, we ensure that $h(x)$ is at least as good as the better of the two individual heuristics for any state. Therefore, it's less likely to overestimate the true cost to reach the goal.

3. Improved Guidance: Using the maximum heuristic, $h(x)$, provides the best guidance among the three options and will likely lead to faster convergence and fewer unnecessary expansions in the search process.