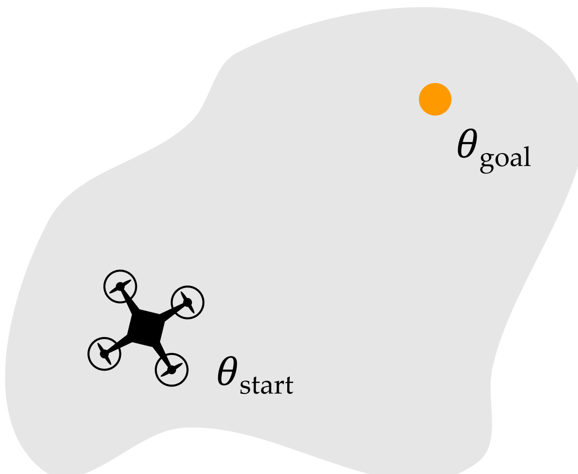
Practice Set 30

Robotics & Automation
Dylan Losey, Virginia Tech

Using your textbook and what we covered in lecture, try solving the following problems. For some problems you may find it convenient to use Matlab (or another programming language of your choice). The solutions are on the next page. **Download the starter code provided with this practice set.**



Let's get a working implementation of trajectory optimization. Consider the 2-DoF environment shown above where θ is the (x, y) coordinates of the drone. You want the drone to plan a trajectory from θ_{start} to θ_{goal} while minimizing the cost function:

$$C(\xi) = \sum_{t=2}^{k} \|\theta^t - \theta^{t-1}\|^2 \tag{1}$$

Problem 1

What will the optimal trajectory look like? Draw a trajectory ξ that minimizes $C(\xi)$ while maintaining the endpoint constraints. Assume the trajectory has $k = 5$ waypoints.

Problem 2

Use the starter code to solve for the optimal trajectory.

- Let $\theta_{start} = [0, 0]^T$ and $\theta_{goal} = [1, 1]^T$
- Set the initial guess $\xi^0 = 0$
- Plot the trajectory for $k = 5$, $k = 10$, and $k = 30$

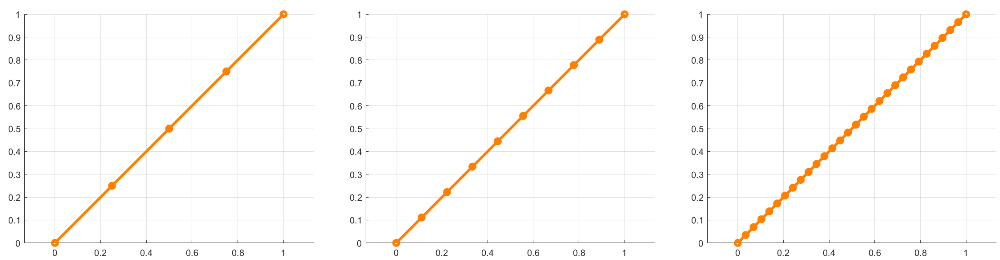
Your plot for $k = 5$ should match the answer you obtained in Problem 1.

Problem 2

Use the starter code to solve for the optimal trajectory.

- Let $\theta_{start} = [0, 0]^T$ and $\theta_{goal} = [1, 1]^T$
- Set the initial guess $\xi^0 = 0$
- Plot the trajectory for $k = 5$, $k = 10$, and $k = 30$

Your plot for $k = 5$ should match the answer you obtained in Problem 1.



See the figure above for the optimal trajectories when $k = 5$, $k = 10$, and $k = 30$. The cost function is minimized when the waypoints are *equally spaced* from start to goal.

To achieve this result you need to make modifications to the provided code. The completed code is shown in the figure below.

```
clear
close all

% start and goal
theta_start = [0;0];
theta_goal = [1;1];

% initial trajectory
n = 2;
k = 5;
xi_0 = zeros(n, k);
xi_0_vec = reshape(xi_0, [], 1);

```

```
% start and goal equality constraints
A = [eye(n) zeros(n, n*(k-1));...
     zeros(n, n*(k-1)), eye(n)];
B = [theta_start; theta_goal];

% nonlinear optimization
options = optimoptions('fmincon','Display','iter',...
     'Algorithm','sqp','MaxFunctionEvaluations',1e5);
xi_star_vec = fmincon(@(xi) cost(xi), xi_0_vec, ...
     [], [], A, B, [], [], [], options);
xi_star = reshape(xi_star_vec, 2, []);

```

```
% plot result
figure
grid on
hold on
axis equal
plot(xi_star(1,:), xi_star(2,:), 'o-',...
     'Color', [1, 0.5, 0], 'LineWidth', 3);

% cost function to minimize
function C = cost(xi)
    xi = reshape(xi, 2, []);
    C = 0;
    for idx = 2:length(xi)
        C = C + norm(xi(:, idx) - xi(:, idx - 1))^2;
    end
end
```