

LAB 7 TRAJECTORY PLANNING

OBJECTIVE

- Emulate the movement of a robotic device from calculated results using forward kinematics
- Estimate the position of individual joints from the calculated results using inverse kinematics
- Generate basic trajectories for a two-link robotic device
- Revise the trajectories for different movements with the consideration of configuration space

INTRODUCTION

In the previous few labs, we have learned basic simulation technique of 2-D robotic systems. In this lab, we will plan the movements of individual joints for specific trajectories. To plan the trajectories of individual joints, both forward and inverse kinematics need to be adopted.

One important issue, you need to keep in mind is that the linear movement is not linear increment due to the rotational movement. To plan the actual trajectory as close as the desired one, the increments of each joint need to be carefully planned to approximate the trajectory. In this lab, we will schedule the rotational movement of the two joints of a 2D robot so that the end effector of the robot is moving along a straight line.

To design the trajectory, geometric configuration and inverse kinematic analysis are required. Assume that the two-link robot is aligned as the following configuration (Fig. 1). In order to have the end effector of the robot moving along the specific line, we need to figure out the angles of θ_1 and θ_2 . As discussed in the lecture, $\theta_1 = 30^\circ$ and θ_2 can have different configuration while keeping the arm located at the identical location. Thus, inverse kinematics is required to determine θ_1 and θ_2 .

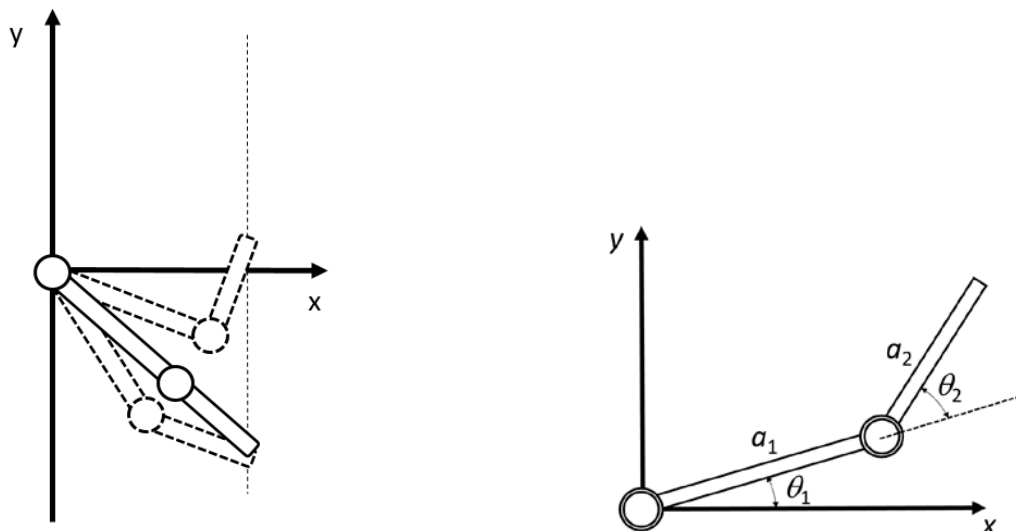


Fig. 1. Configuration of a two-link robot moving along a straight line.

In this lab, we will focus on robot with specific lengths, $a_1 = 3$ ft and $a_2 = 2$ ft, of both links. The robot needs to move along a vertical line $x = 3$ ft. To move the endpoint along the straight

line, both θ_1 and θ_2 need to be divided into several segments. If we start from the lowest point the robot can reach, the angles are $\theta_1 = 53^\circ$ and $\theta_2 = 0^\circ$. We can specify the orientation of the robot to be shown as Figure 2. Then the endpoint of the robot will still be located along the same straight line.

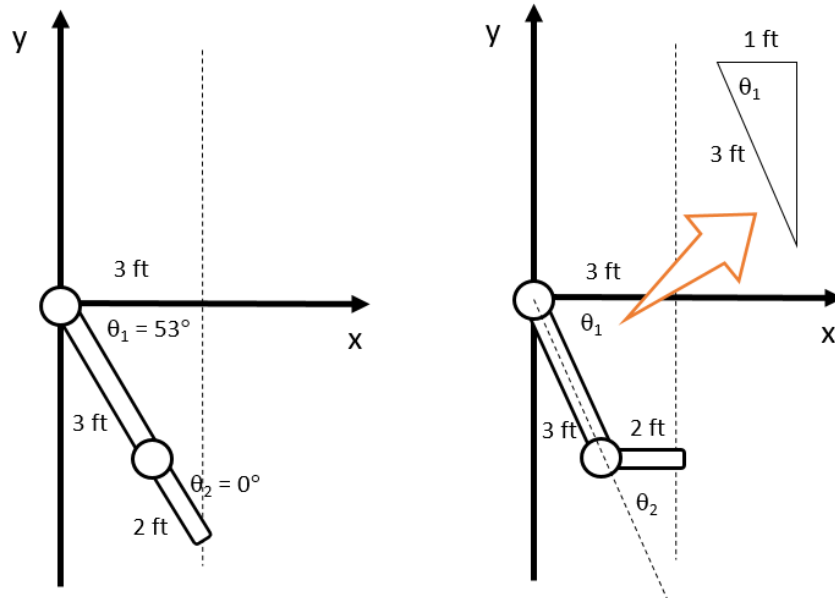


Fig. 2. Movement both joints from one position to another.

The angles of θ_1 and θ_2 can be solved geometrically. With this specific orientation of individual links, it is clear that θ_1 is equal to θ_2 . We can also drive $\theta_1 = 70.5^\circ$. Now, we can start our simulation.

Keep in mind that the endpoint must move along a straight line. Poor scheduling can yield curvilinear trajectory. If you find the movement is not a straight line, you need to divide the segments into smaller ones so that the trajectory can be truly linear. If smaller segment is required, calculate θ_1 and θ_2 accordingly using inverse kinematics.

PROCEDURE

In this lab, we will use the inverse kinematics to identify two functions to generate an animation of a two-link robot. As mentioned in the previous section, the lengths of the two links are $a_1 = 3$ ft and $a_2 = 2$ ft. The desired position of the endpoint of the two-link robot needs to move from $3\mathbf{i} - 4\mathbf{j}$ to $3\mathbf{i} + 4\mathbf{j}$ as a straight line.

I. Single Movement from One Position to another Position

1. Set the orientation of the robot at start point as a straight line to locate endpoint at $3\mathbf{i} - 4\mathbf{j}$. With this configuration, calculate the angles of θ_1 and θ_2 . In the code listed in the next page, replace the value of `theta11` with θ_1 and `theta21` with θ_1 . These two angles are the starting points.
2. Calculate the angles of θ_1 and θ_2 with the second link lies horizontally. Keep in mind that the endpoint of the two-link mechanism must be located at $x = 3$ ft. In the code listed in the next page, replace the value of `theta12` with θ_1 and `theta22` with θ_1 . These two angles are the starting points.

3. Use the following code with the calculated results to simulate the angular movement of the two joints. Observe the movement of the endpoint of the mechanism.

```
clear; clc;

L1 = 3;
L2 = 2;
theta11 =
%=====
p0 = [0 0 0];
theta1 = theta11:(theta12-theta11)/20:theta12;
theta2 = theta21:(theta22-theta21)/20:theta22;

framemax = 20;
M=moviein(framemax);
set(gcf, 'Position',[100 100 640 480]);

for k = 1:20
    T1=RobotConv(theta1(k), 0, L1, 0);
    T2=RobotConv(theta2(k), 0, L2, 0);

    % Forward kinematics
    p0 = [0 0 0];
    p1 = RobotPosition(T1);
    p2 = RobotPosition(T1*T2);

    figure(1)
    X = [p0(1) p1(1) p2(1)];
    Y = [p0(2) p1(2) p2(2)];
    plot(X,Y,'o-')
    axis([-L1-L2 L1+L2 -L1-L2 L1+L2]);
    grid
    M(k) = getframe(gcf);
end
```

4. **Assignment:** Does the endpoint move along a straightline? Describe the movement in your report.

II. Two-Step Movement

1. In some cases, the movement does not fulfill the criterion. You will need to divide the whole stroke into more and smaller segments.
2. Use inverse kinematics to derive θ_2 if θ_1 is given.

	Position 1	Position 2	Position 3
θ_1	53°	70.5°	60°
θ_2			

3. Modify the code in the previous part by adding the following code to the program. Change all the angles accordingly. You need to add one additional for loop to move from position 2 to position 3.

```
theta1 = theta12:(theta13-theta12)/20:theta13;
theta2 = theta22:(theta23-theta22)/20:theta23;

for k = 1:20
    T1=RobotConv(theta1(k), 0, L1, 0);
    T2=RobotConv(theta2(k), 0, L2, 0);

    % Forward kinematics
    p0 = [0 0 0];
    p1 = RobotPosition(T1);
    p2 = RobotPosition(T1*T2);

    figure(1)
    X = [p0(1) p1(1) p2(1)];
    Y = [p0(2) p1(2) p2(2)];
    plot(X,Y,'o-')
    axis([-L1-L2 L1+L2 -L1-L2 L1+L2]);
    grid
    M(k) = getframe(gcf);
end
```

4. Observe the animation of the 2-link robot. Does the movement of the endpoint approximate better to a straight line?

III. Path Planning for the Whole Stroke

1. In this part, you need to plan the trajectory from $3\mathbf{i} - 4\mathbf{j}$ to $3\mathbf{i} + 4\mathbf{j}$ as a straight line. Generate a table with all the planned angles.
2. Write the code and simulate the movement of the two-link mechanism.

DELIVERABLE

- You need to finish a small project. Write a MATLAB code generate the animation with the designed angles.
- Write a Lab report about this lab. You need to include you code and discussion of each assignment in the report.