Faculty of Engineering - Cairo University

Credit Hours System – Freshman Level

Fall - 2021

Cairo University

INTN125

Mechanical Engineering

Project Statement

Submitted to: Dr. Ayman A. Nada

Students: Malak Hamed Mokhtar (1190357) (CCE-C)

Mai Abdelhameed Mohamed (1190365) (CCE-C)

Submission Date: 30/11/2021

➢ Introduction to the manual

This project is made for a robotic arm manipulator of three planar links. The project contains different operations that can be applied on the robotic arm.

When the user runs the program a welcome message appears followed by messages to take in the required parameters. After the user enters the parameters, another message would appear containing the different operations that are offered by our program. The following screenshot is an example:

```
Command Window
Welcome! This is a program containing different modules for a three-link robotic arm
Enter the length of the first link: (in cm) 6
Enter the length of the second link: (in cm) 4
Enter the length of the third link: (in cm) 3
Enter the minimum angle of the first link: (in degrees)
 0
Enter the maximum angle of the first link: (in degrees)
 180
Enter the minimum angle of the second link: (in degrees)
 0
Enter the maximum angle of the second link: (in degrees)
 180
Enter the minimum angle of the third link: (in degrees)
 0
Enter the maximum angle of the third link: (in degrees)
 180
Enter:
 1 for DKPM
 2 for IKPM
 3 for Working Area
 4 for Straight Line Trajectory
 5 for Robot Animation
```

After the user enters the chosen mode of operation, any additional required parameters would be taken and then the output would be displayed (the output of each mode is discussed in the upcoming sections). Following the output, this message would appear in case the user does not want to exit the program and wishes to try different modes of operations on the same set of given input parameters:

```
Do you want to continue? Enter:
1 if yes
0 if no
```

➢ **Contents of the manual:**

In this manual, the user will be able to simulate a 3-link robot arm by:

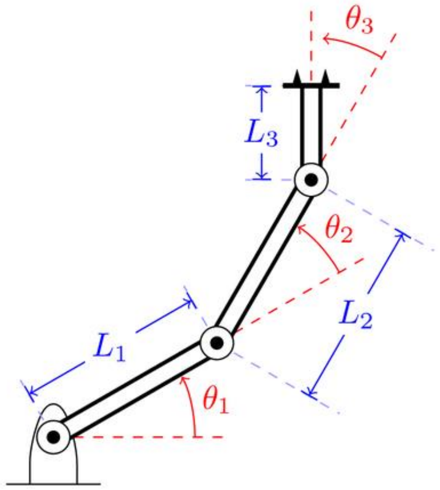
1) Calculating Direct Kinematic Position Model (DKPM)
2) Calculating Inverse Kinematic Position Model (IKPM)
3) Calculating the working area
4) Plotting the working area
5) Generating a straight-line trajectory between two points in the working area
6) Generating a robot animation

1) Direct kinematics Model

Direct Kinematic Position Model (DKPM) refers to the use of the kinematic equations of a robot to compute the position of the end effector from specified values for the joint parameters.

Therefore given θ_1, θ_2 and θ_3 (Figure 1.0) the position of the end effector is determined.

This is calculated using the following two equations:



$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

Figure 1.0

- **Sample input and output:**

```
Command Window
Welcome! This is a program containing different modules for a three-link robotic arm
Enter the length of the first link: (in cm) 6
Enter the length of the second link: (in cm) 5
Enter the length of the third link: (in cm) 3
Enter the minimum angle of the first link: (in degrees)
 0
Enter the maximum angle of the first link: (in degrees)
 180
Enter the minimum angle of the second link: (in degrees)
 0
Enter the maximum angle of the second link: (in degrees)
 180
Enter the minimum angle of the third link: (in degrees)
 0
Enter the maximum angle of the third link: (in degrees)
 180
Enter:
 1 for DKPM
 2 for IKPM
 3 for Working Area
 4 for Straight Line Trajectory
 5 for Robot Animation
1
This is DKPM calculator for a three-link robotic arm of type RRR
Enter the first angle: (in degrees)
 30
Enter the second angle: (in degrees)
 40
Enter the third angle: (in degrees)
 50
The position of the end effector at the given angles is:
 [ 5.406253 , 10.296539 ]
 Do you want to continue? Enter:
 1 if yes
 0 if no
```

2) Inverse kinematics Model

Inverse Kinematics Position Model (IKPM) refers the reverse process that computes the joint parameters that achieve a specified position of the end effector.

Therefore, given the position of the end effector and α (where α is the summation of the three angles), θ_1, θ_2 and θ_3 (Figure 1.0) are determined.

This is calculated using the following relations:

$$\overline{X} = x - l_3 \cos(\alpha)$$

$$\overline{Y} = y - l_3 \sin(\alpha)$$

$$\theta_2 = \cos^{-1}\left(\frac{\overline{X}^2 + \overline{Y}^2 - l_1^2 - l_2^2}{2(l_1 * l_2)}\right)$$

$$\begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} = \frac{1}{\overline{X}^2 + \overline{Y}^2} \begin{bmatrix} l_1 + l_2\cos(\theta_2) & l_2\sin(\theta_2) \\ -l_2\sin(\theta_2) & l_1 + l_2\cos(\theta_2) \end{bmatrix} \begin{bmatrix} \overline{X} \\ \overline{Y} \end{bmatrix}$$

$$\theta_3 = \alpha - (\theta_1 + \theta_2)$$

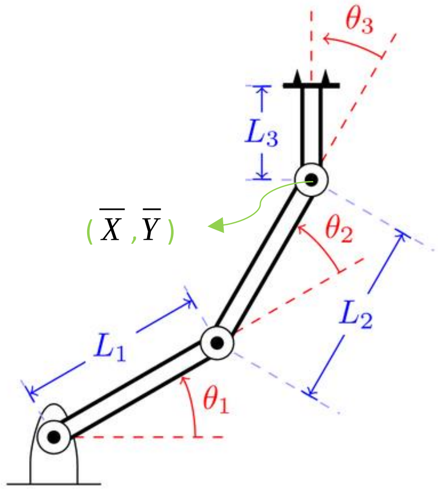


Figure 1.1

However, this method would be too difficult to compute on the software. Hence, we have developed a method where a random phi is generated within the ranges of Theta1, Theta2 and Theta3.

- **Sample Input and Output:**

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Welcome! This is a program containing different modules for a three-link robotic arm
Enter the length of the first link: 6
Enter the length of the second link: 4
Enter the length of the third link: 3
Enter the minimum angle of the first link: (in degrees)
 0
Enter the maximum angle of the first link: (in degrees)
 180
Enter the minimum angle of the second link: (in degrees)
 0
Enter the maximum angle of the second link: (in degrees)
 180
Enter the minimum angle of the third link: (in degrees)
 0
Enter the maximum angle of the third link: (in degrees)
 180
Enter:
 1 for DKPM
 2 for IKPM
 3 for Working Area
 4 for Straight Line Trajectory
 5 for Robot Animation
2
Enter the value for x-coordinate: 5.406
Enter the value for y-coordinate: 10.296
The angles of the end effector at the given coordinates are:
 [ 33.452767 , 53.065102 , 0.482130 ]
 Do you want to continue? Enter:
 1 if yes
 0 if no
```

3) Calculation of working area using Green's theorem

The Working Area of a robot manipulator is defined as the set of points that can be reached by its end-effector.

Therefore, given the minimum and maximum angles of each link, the working area covered is calculated.

- **Sample Input and Output:**

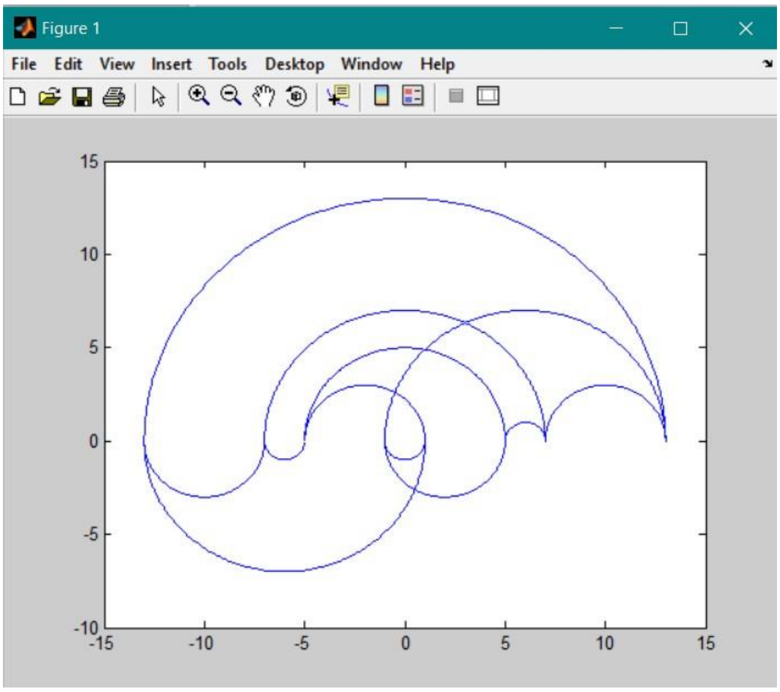
  The following screenshot shows the sample input and the working area calculation output:

```
Command Window
Welcome! This is a program containing different modules for a three-link robotic arm
Enter the length of the first link: (in cm) 6
Enter the length of the second link: (in cm) 4
Enter the length of the third link: (in cm) 3
Enter the minimum angle of the first link: (in degrees)
 0
Enter the maximum angle of the first link: (in degrees)
 180
Enter the minimum angle of the second link: (in degrees)
 0
Enter the maximum angle of the second link: (in degrees)
 180
Enter the minimum angle of the third link: (in degrees)
 0
Enter the maximum angle of the third link: (in degrees)
 180
Enter:
 1 for DKPM
 2 for IKPM
 3 for Working Area
 4 for Straight Line Trajectory
 5 for Robot Animation
3
This is the calculation and plotting of the Working Area for a three-link robotic arm of type RRR
The working area= 331.492691
```

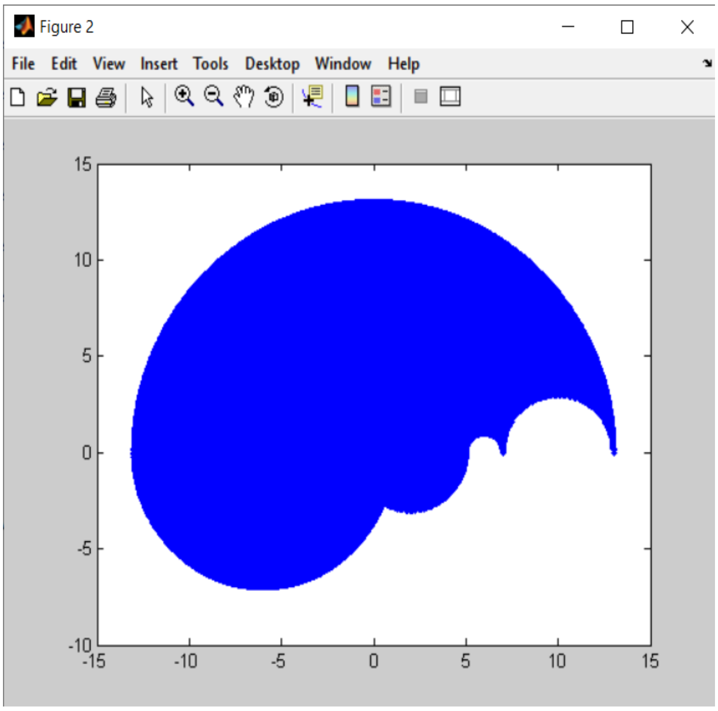
4) Plotting the working area

In this option, the user will be able to view the working area of the robot arm.

1) Figure 1 is displayed on the screen for the user which presents the different paths the end effector might take:



2) Figure 2 is also displayed for the user, this figure shows the working area of the robot manipulator (filled in blue).
   - ✓ Note that any point outside this working area (in blue) cannot be reached by the robot.

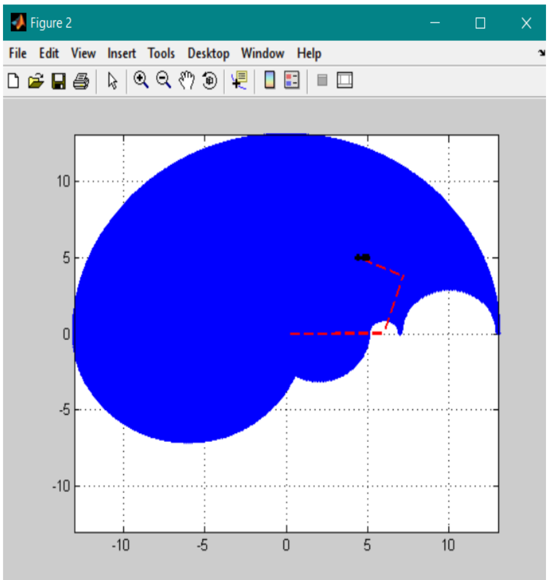


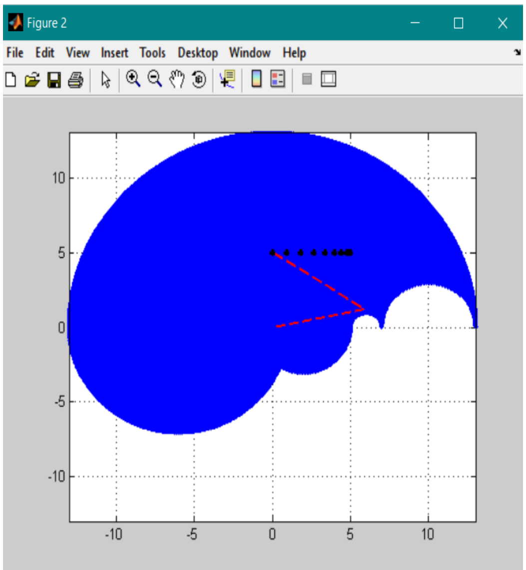
5) Generation of straight-line trajectory between two points in the working area

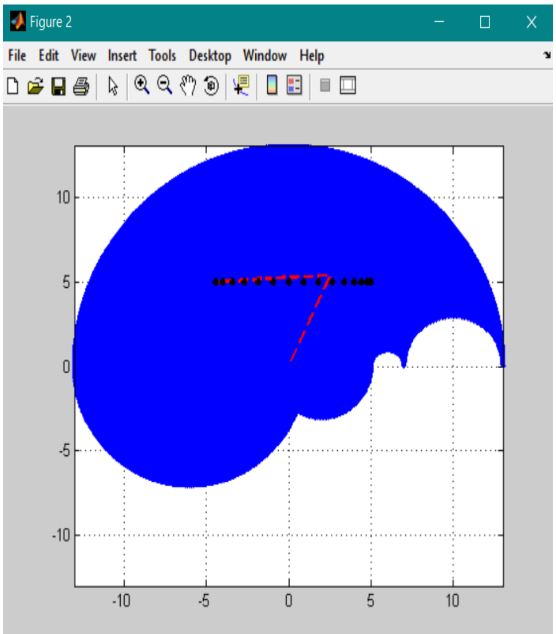
This simulation option allows the user to view how the robot arm would move on a straight-line between two given points, initial point Po and final point Pf.

As show in the figures below, the straight-line is shown as a dotted line and the robot arm is moving along this line.

- ✓ Note that any points chosen outside the blue filled region is invalid as the robot arm will not be able to reach this point.







6) Robot animation

This last simulation option gives the user the overall movement of the robot arm along all paths. Hence, it shows how the robot would move in reality along the allowed region using the given dimensions and angles.