INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

(Company No. 101067-P)

Garden of Knowledge and Virtue

PREMIER INTERNATIONAL ISLAMIC RESEARCH UNIVERSITY

KULLIYYAH OF ENGINEERING
DEPARTMENT OF MECHATRONICS ENGINEERING

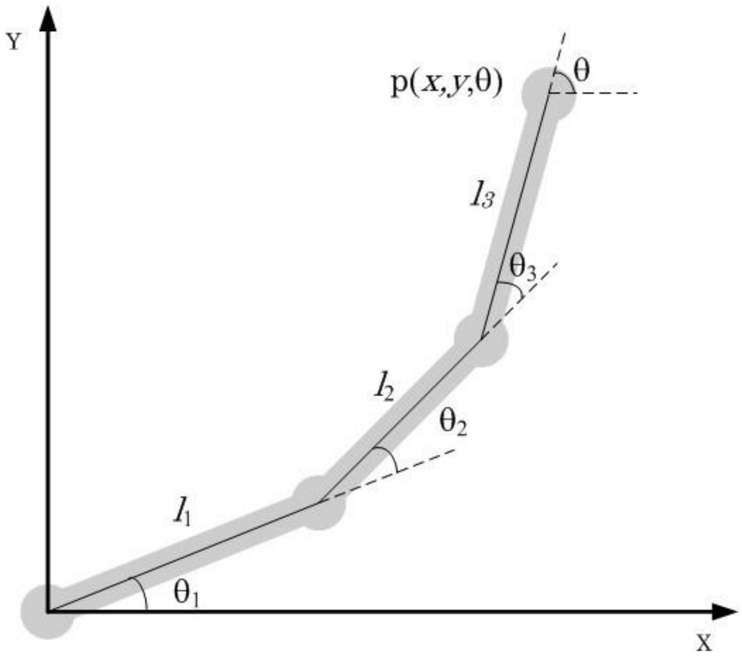
MCTE 4352
ROBOTICS

Abdulrhamn Ghnem

1625999

Sec1

DR. TANVEER SALEH



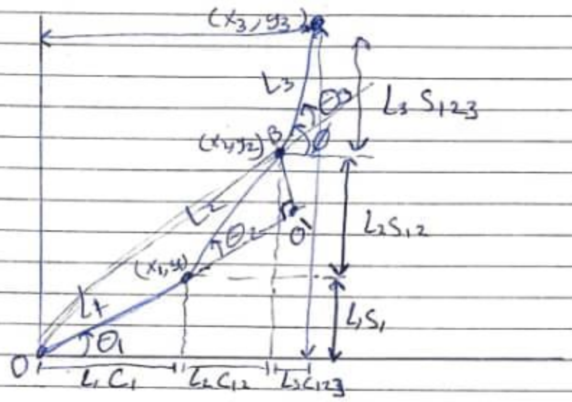
DH parameters:

Link	a_i	α_i	d_i	θ_i
1	l_1	**0**	**0**	Θ_1
2	l_2	**0**	**0**	Θ_2
3	l_3	**0**	**0**	Θ_3

Forward Kinematic equation: using geometric.

$x = l1c1 + l2c12 + l3c123$

$y = l1S1 + l2S12 + l3S123$



- Forward Kinematics

$$x_3 = L_1 C_1 + L_2 C_{12} + L_3 C_{123}$$

$$y_3 = L_1 S_1 + L_2 S_{12} + L_3 S_{123}$$

$$\phi = \theta_1 + \theta_2 + \theta_3$$

- Inverse Kinematics

given $(x_3, y_3, L_1, L_2, L_3 \& \phi)$
need to find $(\theta_1, \theta_2, \theta_3)$?

1) $x_2 = x_3 - L_3 C\phi$
$y_2 = y_3 - L_3 S\phi$

$\Big\{$ rewrite (x_2) in terms of (x_3)

$$C\theta_2 = \frac{x_2^2 + y_2^2 - L_1^2 - L_2^2}{2 L_1 L_2}$$

$$x^2 + y^2 = OB^2$$

$$OB^2 = O'B^2 + OO'^2$$

$$(L_2 S_2)^2 + (L_1 + L_2 C_2)^2$$

$$x^2 + y^2 = L_2^2 S_2^2 + L_1^2 + L_2^2 C_2^2 + 2 L_1 L_2 C_2$$

$$2 L_1 L_2 C_2 = x^2 + y^2 - L_1^2 - L_2^2$$

$$C_2 = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2 L_1 L_2}$$

No: ______ (3) Date: ______

write x_2 in terms of (x_1)

$x_2 = L_1C_1 + L_2C_{12}$ $\quad y_2 = L_1S_1 + L_2S_{12}$

$\boxed{\cos(x+y) = \cos x \cos y + \sin x \sin y}$

$x_2 = L_1C_1 + L_2[C_1C_2 + S_1S_2]$

$= L_1C_1 + L_2C_1C_2 + L_2S_1S_2$

$x_2 = C_1(L_1 + L_2C_2) + S_1(L_2S_2) \longrightarrow (1)$

$\boxed{\sin(x \pm y) = \sin x \cos y \pm \cos x \sin y}$

$y_2 = L_1S_1 + L_2[S_1 \cdot C_2 + C_1S_2]$

$y_2 = L_1S_1 + L_2S_1C_2 + L_2C_1S_2$

$= C_1(L_2S_2) + S_1(L_1 + L_2C_2) \longrightarrow (2)$

$$C_1 = \frac{(L_1 + L_2C_2)x_2 + L_2S_2y_2}{x_2^2 + y_2^2}$$

$$S_1 = \frac{(L_1 + L_2C_2)y_2 - L_1S_2x_2}{x_2^2 + y_2^2}$$

$\tan\theta_1 = \frac{S_1}{C_1} \Rightarrow \theta_1 = \tan^{-1}\left(\frac{S_1}{C_1}\right)$

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

after comparing the results from my calculation and from the simulation code, it was different, and this because inverse kinematics do not have unique solution.

And the function used in the code is ikine which finds the solution numerically.

Example:

Matlab: Θ_{2}= 0.7109

Calculations: Θ_{2}= -0.6537

Code explanation

```
L1=10; L2=20; L3=8; % define the length of the robot
L(1) = Link([0 0 L1 0]); %D-H parameter for link 1
L(2) = Link([0 0 L2 0]);%D-H parameter for link 1
L(3) = Link([0 0 L3 0]);%D-H parameter for link 1
ThreeLink = SerialLink(L);% construct the robot
ThreeLink.name = 'Planar3R'; % give it a name
ThreeLink.base=[0 -10 0]; % define the base
```

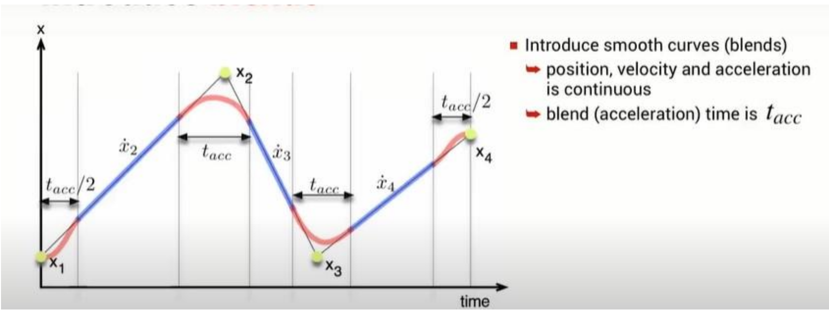
Defining the length of the robot and constructing the D-H table. The main point in this code is to make sure that the length of the robot is able to reach the farthest point on your scale and if not the you will get an error that the inverse kinematics function can't converge which means there is no solution.

```
a = xlsread('GHNEM1.xlsx'); % read the points from an excel file
```

In this part I'm just reading the points from an excel file, but here we should be careful, our robot is a planner robot so all the points are in X and Y , but the functions in the toolbox need to define X,Y,Z so in our case Z = 0 for all our points.

```
masd=mstraj(a,[6,6,0],[],a(1,:),0.1,0); % use the mstraj function to find the trajectory
```

The mstraj function helps to define the via path which is (a "my name points") and what it does that it uses the concept of multi-segment , and what that means , it creates path with a points near as possible to the via path , and that because to have a constant speed and accelerate in some points. As shown in the figure below.




```
trsl=transl(masd); % find the x-y-z of the trajectory
```

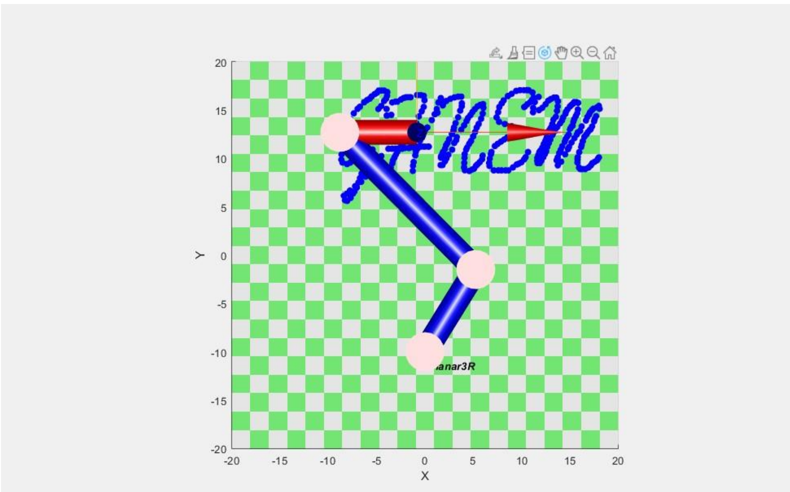
in this code the function transl just put the xyz points we got from the mstraj function in a 4x4 matric , because the ikine function takes only 4x4 as an argument

```
q1=ThreeLink.ikine(transl(masd),[0 0 0],'mask',[1 1 0 0 0 1]); % find the inverse kinematics
```

the ikine function takes 4x4 matric and we need to define the [0 0 0] as second argument and that means the initial position and for any robot which less than 6DOF we need to use a mask and in the mask we define the [x y z rotx roty rotz] in our case we have a motion in x and y so that I put 1 in those two positions and since we don't have a motion in z I put 0 and for the rotation we zero rotation in x and y but we have a rotation z which equal to 1.

```
ThreeLink.plot(q1,'loop','workspace',[-20 20 -20 20 0 20]); % plot the robot giving it the inverse kinematics
```

In this code we just plot the robot using plot function and we should pass the theta we got from the inverse kinematics and it advisable to choose a scaled workspace.



Github : [abdulrhman-AIG/3-DoF-planar-robot-Simulation (github.com)]