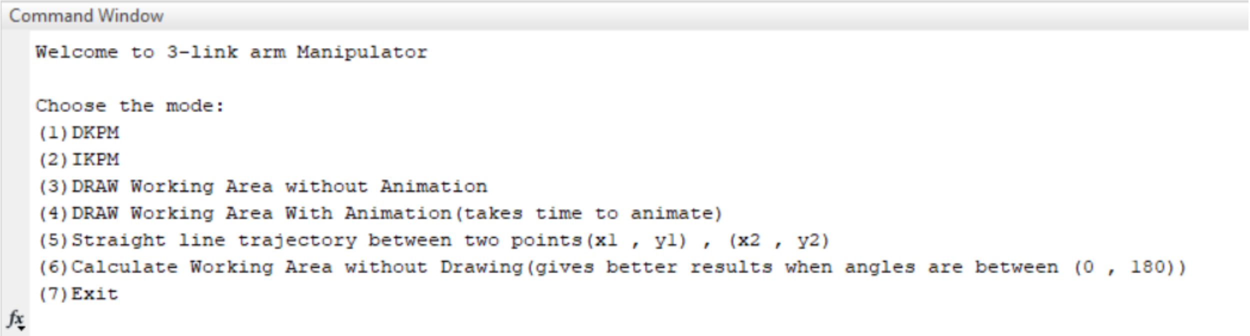
Mechanical Engineering
Project Report

Name: Ahmed Mohamed Saad Hussein

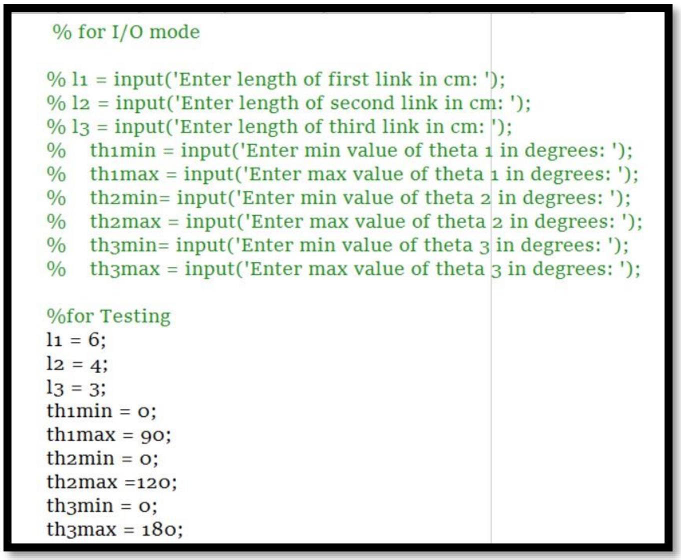
ID: 1190184



```
Welcome to 3-link arm Manipulator

Choose the mode:
(1)DKPM
(2)IKPM
(3)DRAW Working Area without Animation
(4)DRAW Working Area With Animation(takes time to animate)
(5)Straight line trajectory between two points(x1 , y1) , (x2 , y2)
(6)Calculate Working Area without Drawing(gives better results when angles are between (0 , 180))
(7)Exit
```

1- Main Program

The main is an endless loop where you can change the mode you want infinite number of times and if you want to exit press 7.



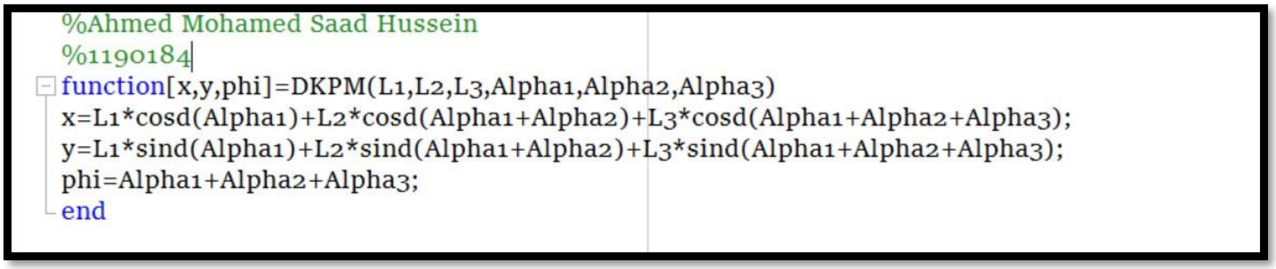
```
 % for I/O mode

% l1 = input('Enter length of first link in cm: ');
% l2 = input('Enter length of second link in cm: ');
% l3 = input('Enter length of third link in cm: ');
%    th1min = input('Enter min value of theta 1 in degrees: ');
%    th1max = input('Enter max value of theta 1 in degrees: ');
%    th2min= input('Enter min value of theta 2 in degrees: ');
%    th2max = input('Enter max value of theta 2 in degrees: ');
%    th3min= input('Enter min value of theta 3 in degrees: ');
%    th3max = input('Enter max value of theta 3 in degrees: ');

%for Testing
l1 = 6;
l2 = 4;
l3 = 3;
th1min = 0;
th1max = 90;
th2min = 0;
th2max =120;
th3min = 0;
th3max = 180;
```

If you want to enter the length of the links and the range of angles you should uncomment the I/O mode commented section and comment the Testing mode section or else use the Testing Mode .

All the following tests are done at the parameters in the picture above.

2-Functions

2-1 DKPM

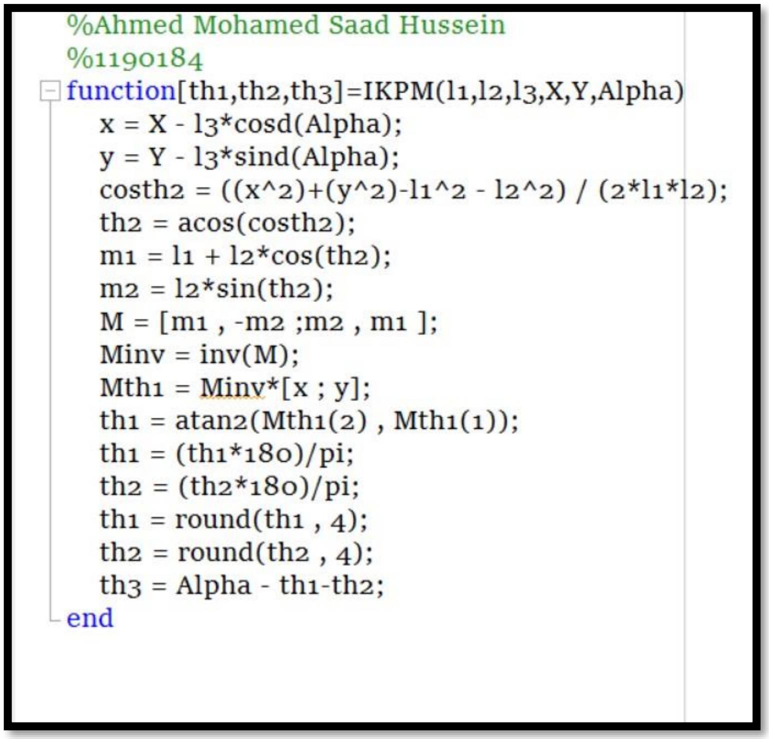


```
%Ahmed Mohamed Saad Hussein
%1190184
function[x,y,phi]=DKPM(L1,L2,L3,Alpha1,Alpha2,Alpha3)
x=L1*cosd(Alpha1)+L2*cosd(Alpha1+Alpha2)+L3*cosd(Alpha1+Alpha2+Alpha3);
y=L1*sind(Alpha1)+L2*sind(Alpha1+Alpha2)+L3*sind(Alpha1+Alpha2+Alpha3);
phi=Alpha1+Alpha2+Alpha3;
end
```

Basic Function that uses The model we studied before takes the 3 angles of the 3 link arm and return x , y , phi(sum of three angles).

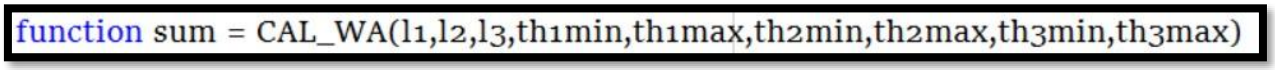
2-2 IKPM

Using the IKPM model that was studied before (using matrices) , This Function takes X , Y , phi and returns the three angles.



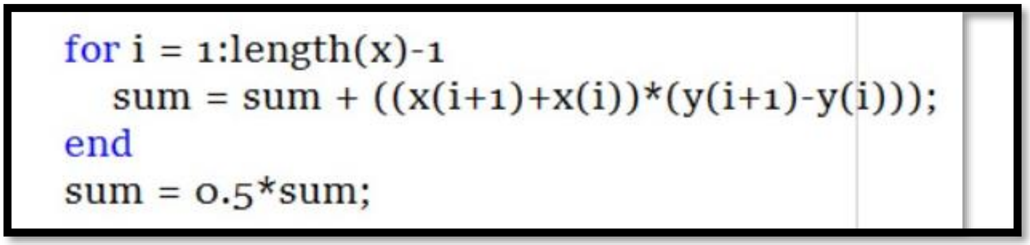
```
%Ahmed Mohamed Saad Hussein
%1190184
function[th1,th2,th3]=IKPM(l1,l2,l3,X,Y,Alpha)
    x = X - l3*cosd(Alpha);
    y = Y - l3*sind(Alpha);
    costh2 = ((x^2)+(y^2)-l1^2 - l2^2) / (2*l1*l2);
    th2 = acos(costh2);
    m1 = l1 + l2*cos(th2);
    m2 = l2*sin(th2);
    M = [m1 , -m2 ;m2 , m1 ];
    Minv = inv(M);
    Mth1 = Minv*[x ; y];
    th1 = atan2(Mth1(2) , Mth1(1));
    th1 = (th1*180)/pi;
    th2 = (th2*180)/pi;
    th1 = round(th1 , 4);
    th2 = round(th2 , 4);
    th3 = Alpha - th1-th2;
end
```

2-3 CAL_WA (calculation of working area)



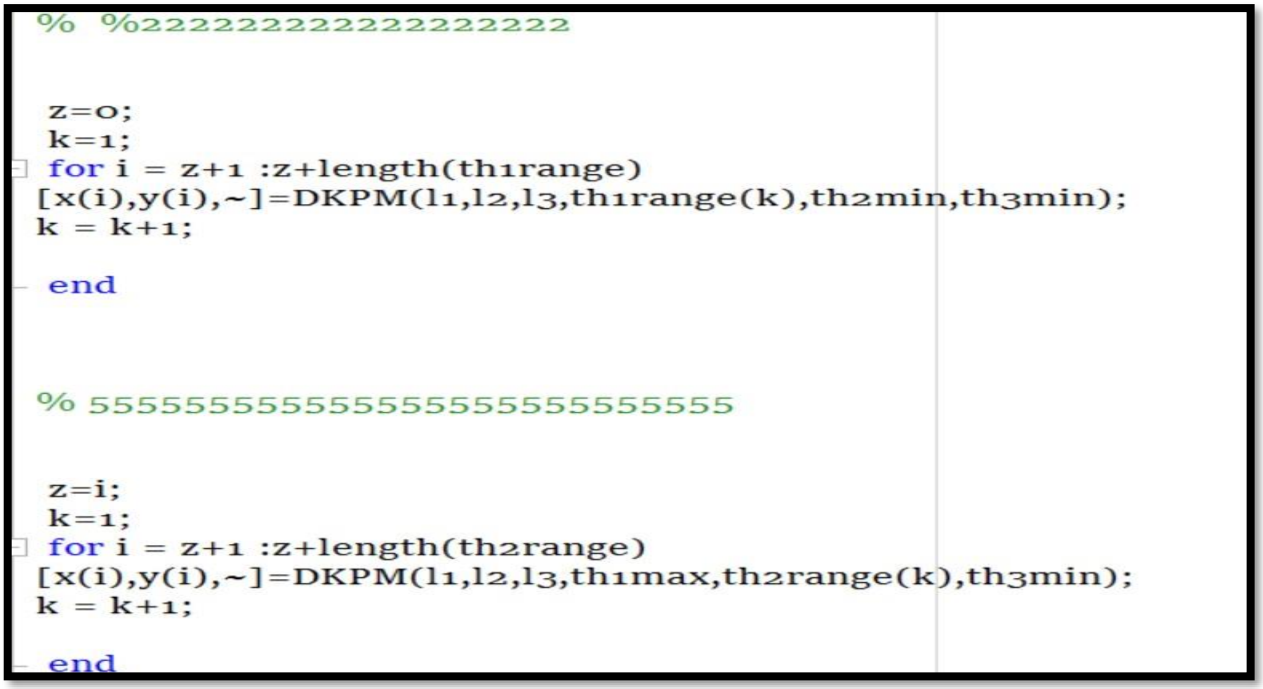
```
function sum = CAL_WA(l1,l2,l3,th1min,th1max,th2min,th2max,th3min,th3max)
```

Uses Green Theorem by calculating the points that makes the outer frame of the working area and orders them in counter-clockwise order then use the green theorem formula given at the lecture.



```
for i = 1:length(x)-1
   sum = sum + ((x(i+1)+x(i))*(y(i+1)-y(i)));
end
sum = 0.5*sum;
```

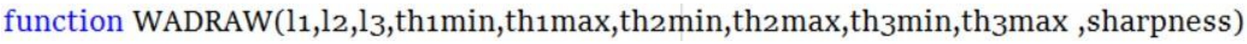
Outer points are calculated as paths each one has an order the figure below shows 2 paths’ points being calculated.



```
% %222222222222222222


 z=0;
 k=1;
 for i = z+1 :z+length(th1range)
[x(i),y(i),~]=DKPM(l1,l2,l3,th1range(k),th2min,th3min);
k = k+1;

 end



% 5555555555555555555555555555


 z=i;
 k=1;
 for i = z+1 :z+length(th2range)
[x(i),y(i),~]=DKPM(l1,l2,l3,th1max,th2range(k),th3min);
k = k+1;

 end
```

NB#: Since phi not used there I put (~) instead in DKPM

2-4 WADRAW (Drawing of working area)

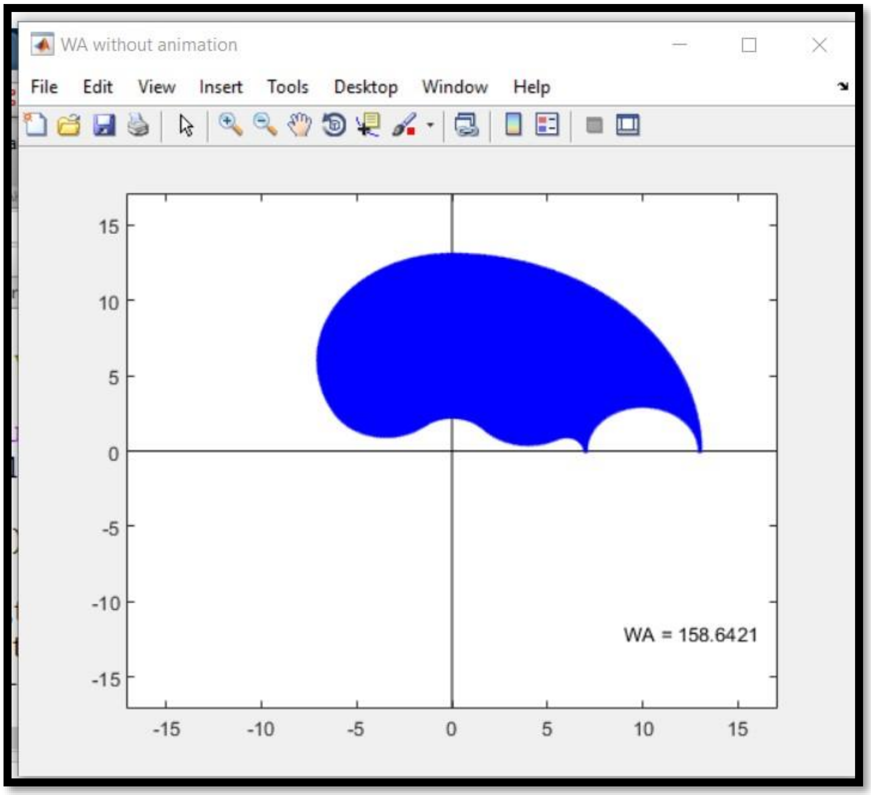


```
function WADRAW(l1,l2,l3,th1min,th1max,th2min,th2max,th3min,th3max ,sharpness)
```

Estimated working area is printed on the graph when its drawn.

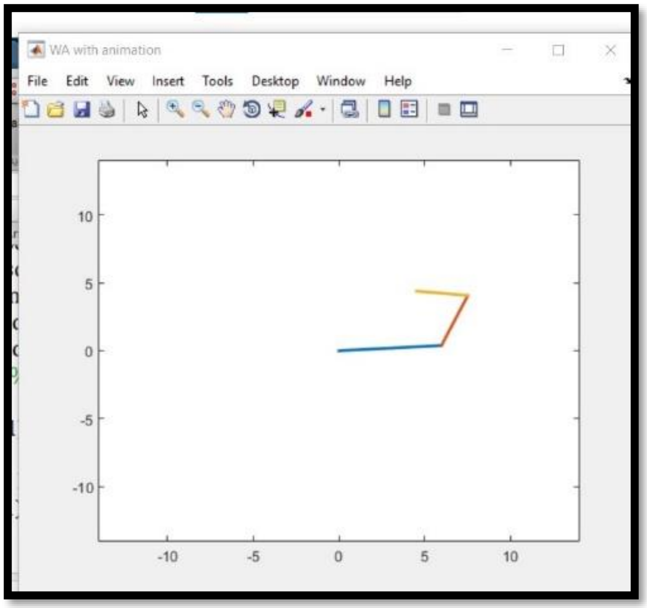
The Sharpness parameter indicates the no of points used for each range of the 3 angles increasing it makes sharper Working Area But takes more time.

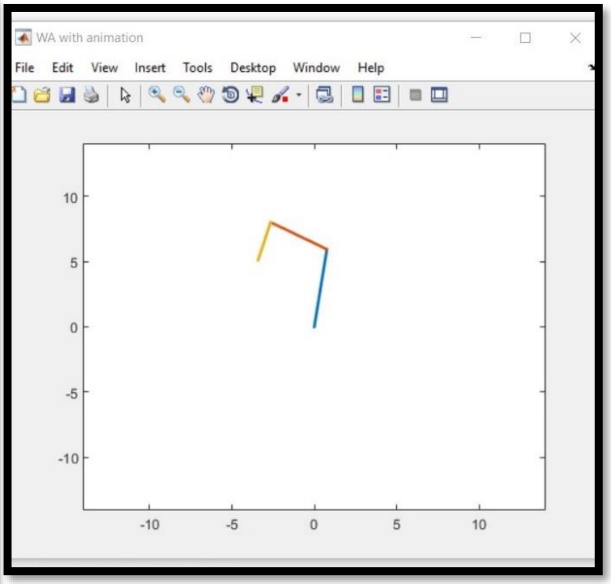
(For the clearest WA -> sharpness = 100, For the fastest WA -> sharpness = 25)



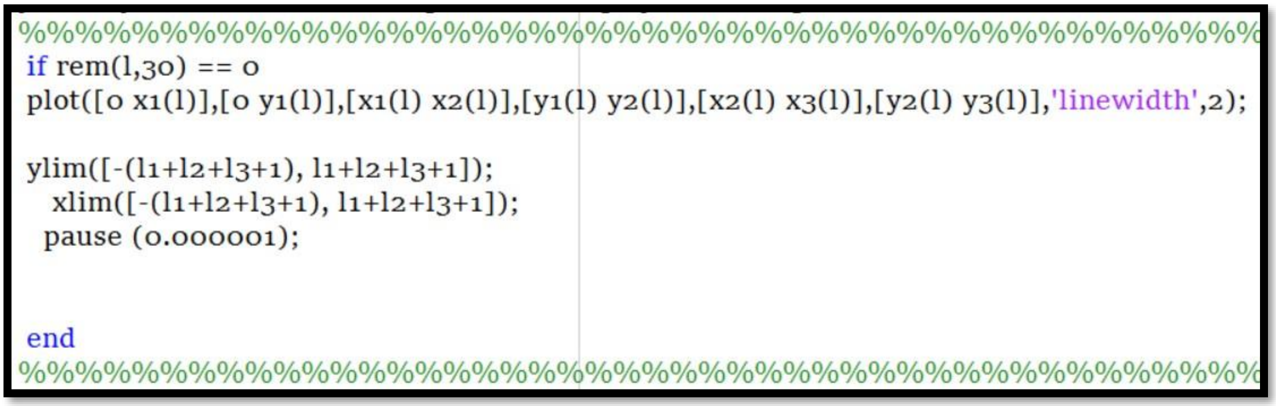
2-5 WAANIM (Drawing of working area with moving 3 link arm)

Same as WADRAW but first it draws a moving 3 link arm that scans the working area.





A piece of code is added

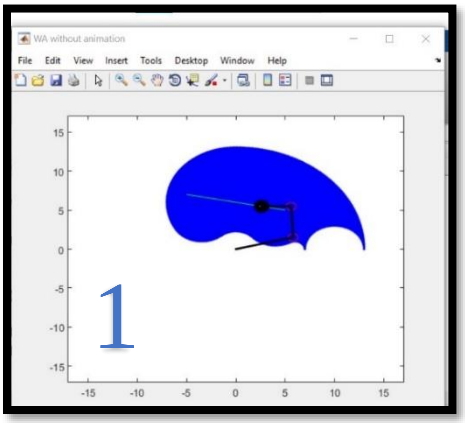


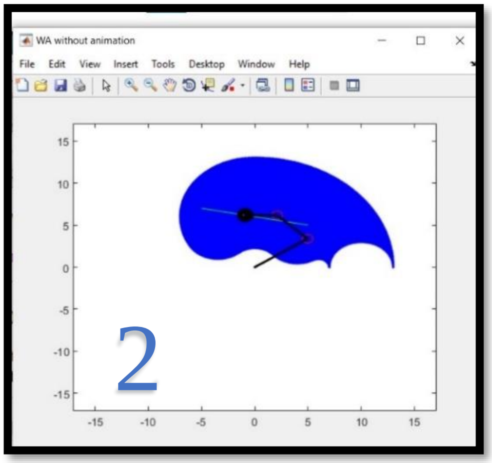
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 if rem(l,30) == 0
 plot([0 x1(l)],[0 y1(l)],[x1(l) x2(l)],[y1(l) y2(l)],[x2(l) x3(l)],[y2(l) y3(l)],'linewidth',2);

 ylim([-(l1+l2+l3+1), l1+l2+l3+1]);
   xlim([-(l1+l2+l3+1), l1+l2+l3+1]);
  pause (0.000001);


 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

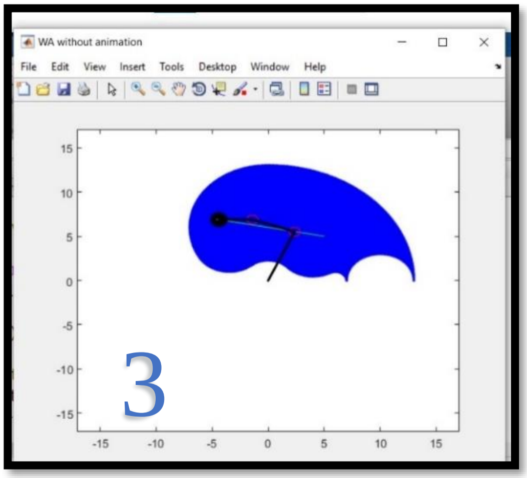
The value 30 **(draws one time every 30 point)** in the rem function is added for testing reasons to speed up the process of simulation while testing but it can be reduced to 1 to draw point by point.

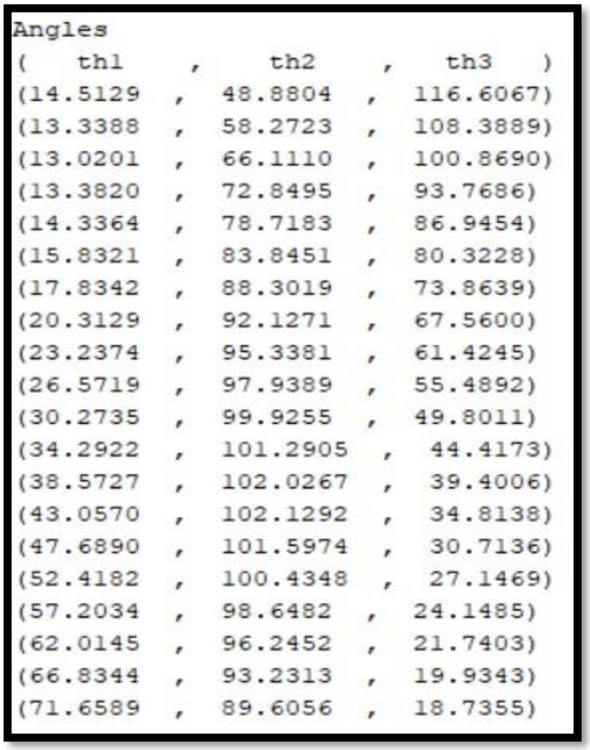
2-6 LinkTraj (Link Trajectory)

Used to draw straight line Trajectory between 2 points on working area and the arm walks between them giving the readings of the angles in the command window.







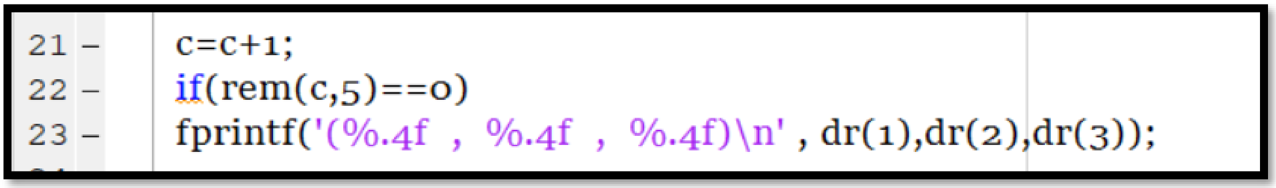


```
Angles
(   th1      ,      th2      ,     th3    )
(14.5129   ,   48.8804   ,   116.6067)
(13.3388   ,   58.2723   ,   108.3889)
(13.0201   ,   66.1110   ,   100.8690)
(13.3820   ,   72.8495   ,   93.7686)
(14.3364   ,   78.7183   ,   86.9454)
(15.8321   ,   83.8451   ,   80.3228)
(17.8342   ,   88.3019   ,   73.8639)
(20.3129   ,   92.1271   ,   67.5600)
(23.2374   ,   95.3381   ,   61.4245)
(26.5719   ,   97.9389   ,   55.4892)
(30.2735   ,   99.9255   ,   49.8011)
(34.2922   ,   101.2905   ,   44.4173)
(38.5727   ,   102.0267   ,   39.4006)
(43.0570   ,   102.1292   ,   34.8138)
(47.6890   ,   101.5974   ,   30.7136)
(52.4182   ,   100.4348   ,   27.1469)
(57.2034   ,   98.6482   ,   24.1485)
(62.0145   ,   96.2452   ,   21.7403)
(66.8344   ,   93.2313   ,   19.9343)
(71.6589   ,   89.6056   ,   18.7355)
```

Angles written in command window

#NB: I used MATLAB version R2015a.

In line 22: The value 5 (Draws only one time each 5 points) is added for faster testing but can be changed to 1 to be more accurate.



```
21 -     c=c+1;
22 -     if(rem(c,5)==0)
23 -     fprintf('(%.4f  ,  %.4f  ,  %.4f)\n' , dr(1),dr(2),dr(3));
```