Robot Programming

Lecture #8

ECE 761: Robotics

Class taught at North Dakota State University
Department of Electrical and Computer Engineering

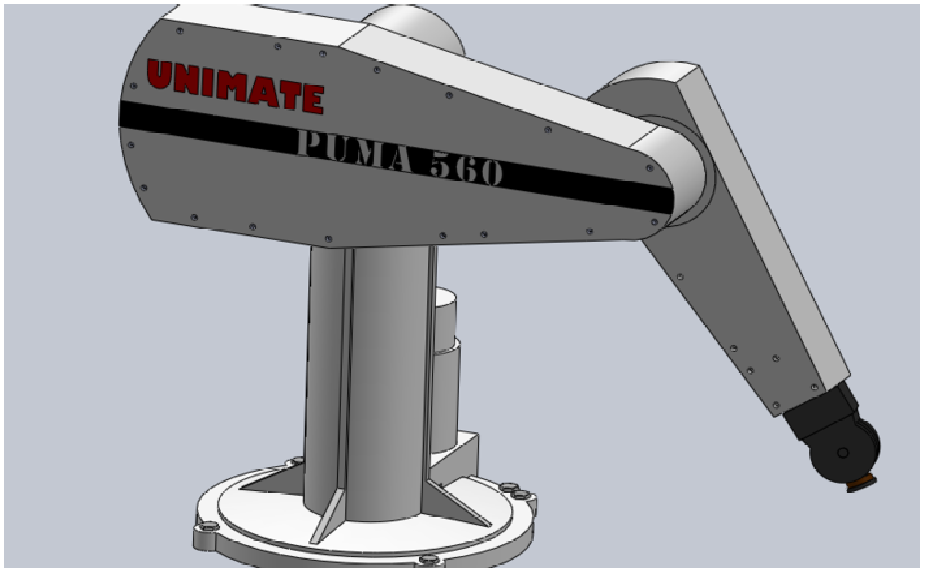
Please visit www.BisonAcademy.com for corresponding lecture notes, homework sets, and solutions.

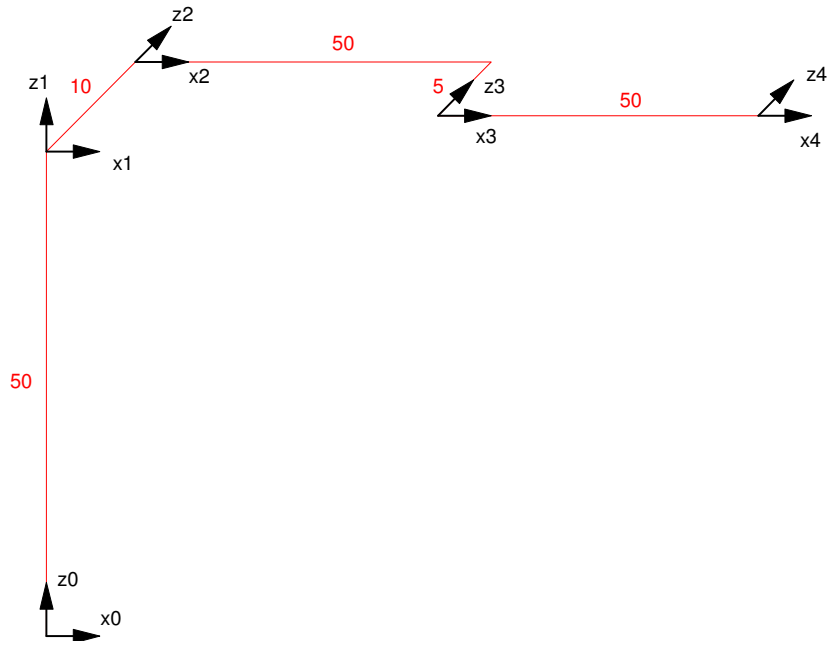
Robot Programming

So far, we have

- Forward kinematics (find the tip position given angles)
- Inverse kinematics (find the angles given the tip position)

Write program to trace out a specified shape.





Useful Routines

```
[Q] = MoveTo(P0, P1, T)
```

- Go from point P0 to point P1 in T seconds in a straight line. Use cosine interpolation so that the initial and final velocities are zero. The function returns the final position of the robot (Pf) and the joint angles every 0.01 second.

```
[Q] = Circle(P0, Pc, T)
```

- Draw a circle, starting at point P0, with a center of the circle at Pc, in T seconds.

```
[Q] = ArcXY(P0, Pc, angle, T)
```

- Draw an arc in the XY plane, starting at P0, with the center of the circle at Pc, at angle radians, in T seconds.

```
[Q] = ArcXY2(P1, P2, P3, T)
```

- Draw an arc in the XY plane, starting at point P1, passing through point P2, ending at point P3, in T seconds.

MoveTo:

Use interpolation to find points from P0 to P1 every 0.01 second

$$P_{tip} = (1-a)P_0 + (a)P_1$$

where

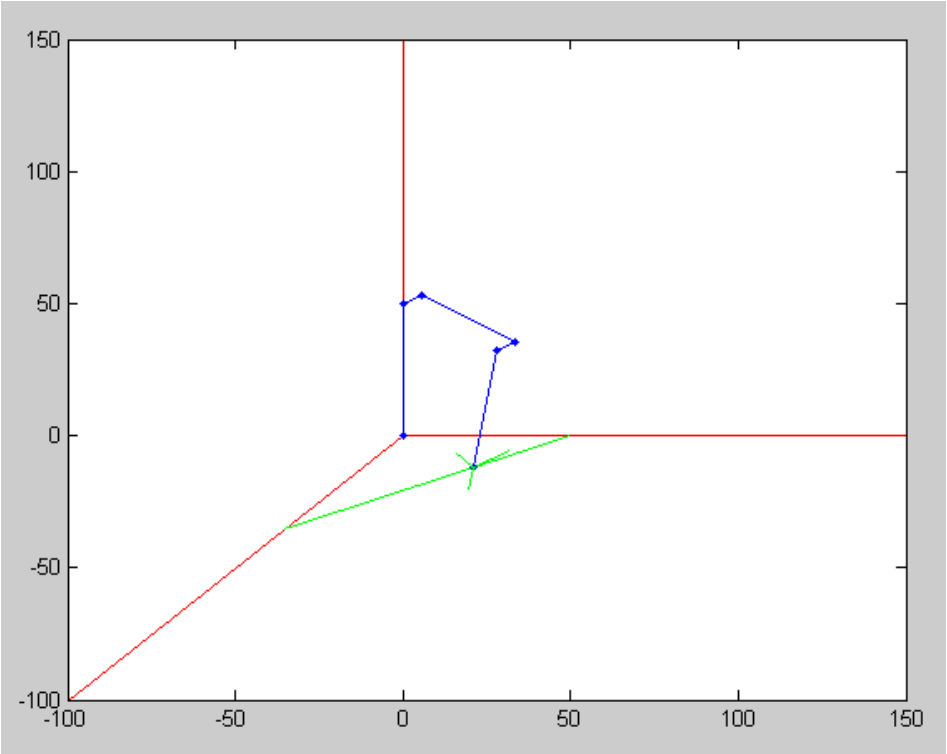
$$0 < a < 1$$

Code:

```
function [Q] = MoveTo(P0, P1, T)
% move from point P0 to P1 in T seconds

t = [0:0.01:T];
a = (1 - cos(pi*t/T))/2;

TIP = P0 * (1-a)  + P1 * a;

end
```

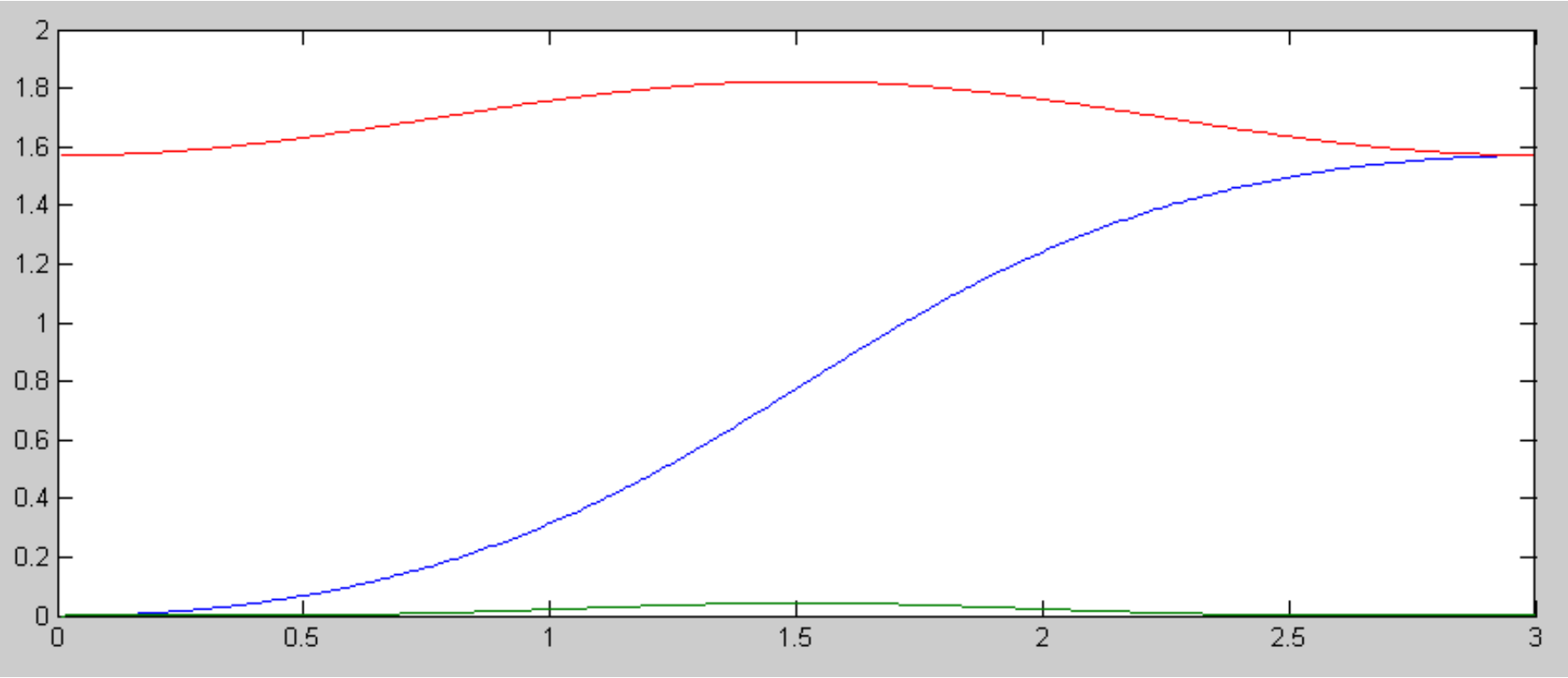
Example: Draw a line from [50,0,0]' to [0,50,0]' in 3 seconds.

```
P1 = [50,0,0]';
P2 = [0,50,0]';

Q = MoveTo(P1, P2, 3);
```



To plot the joint angles during this motion, use

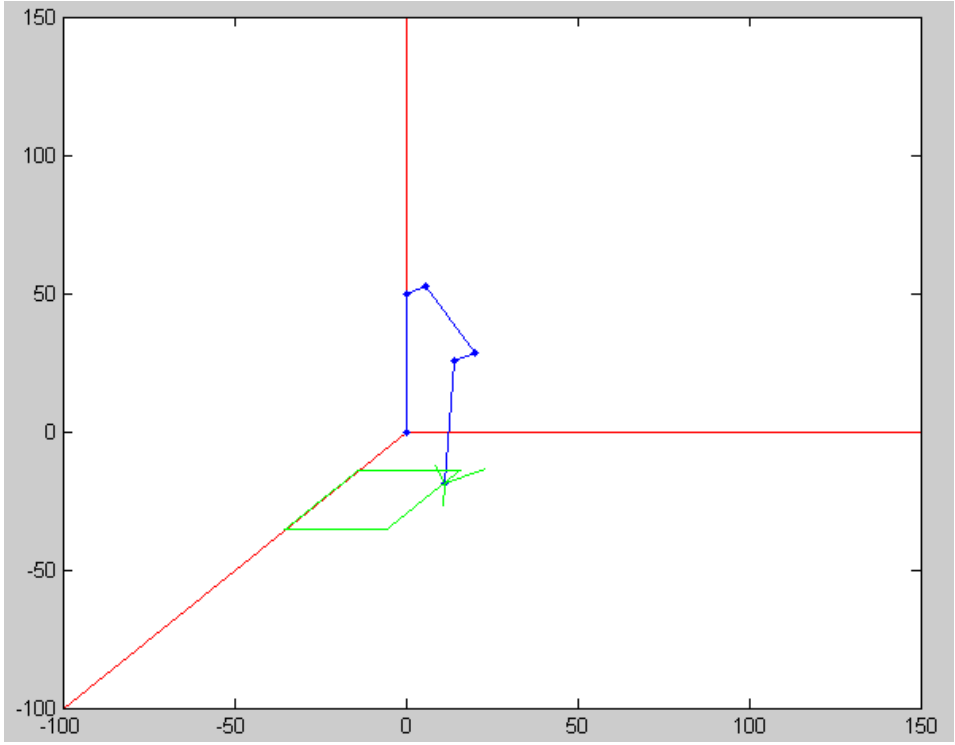
```
t = [1:length(Q)]' * 0.01;
clf
plot(t,Q')
```



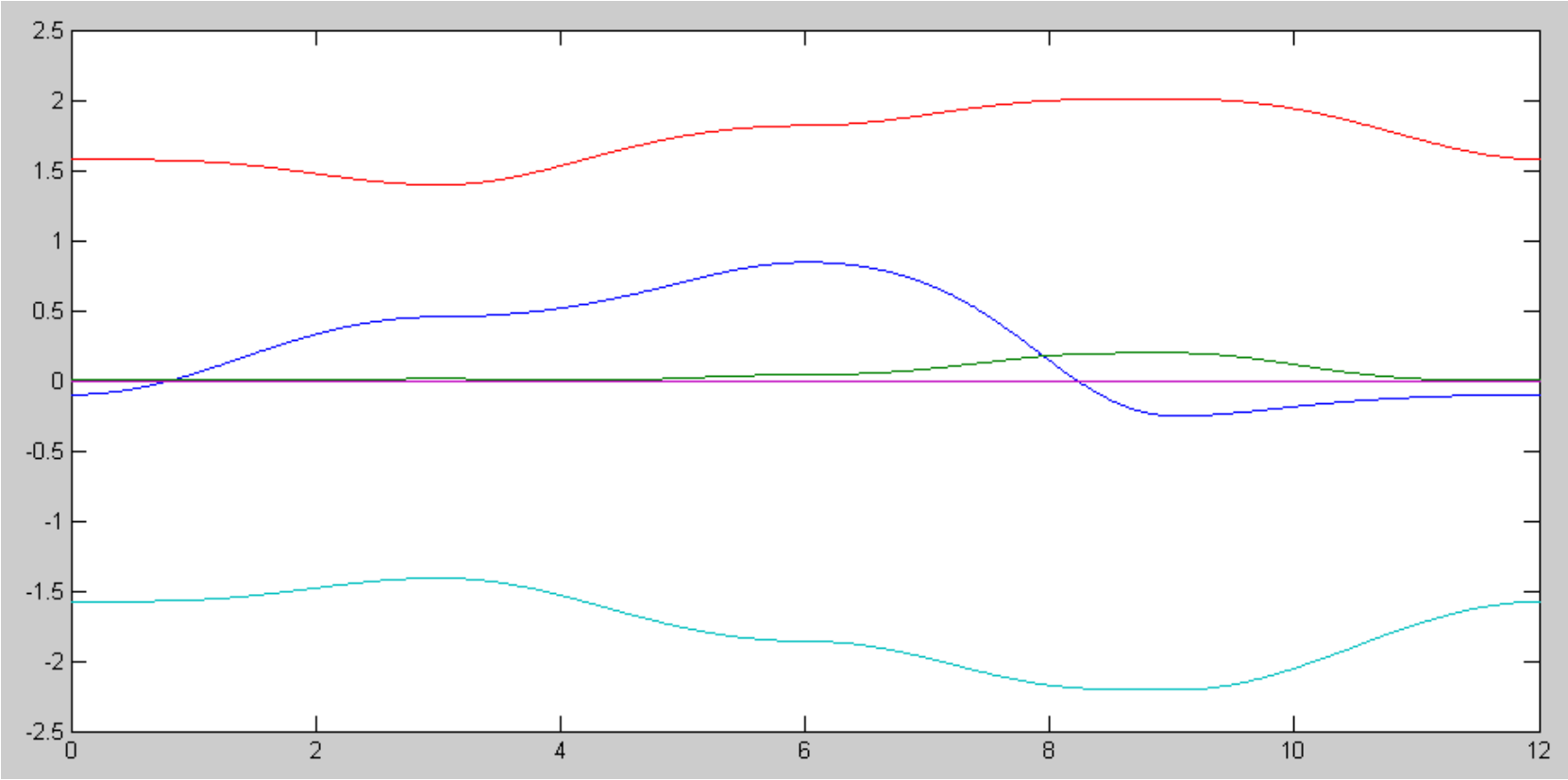
Joint Angles during the MoveTo command (cosine interpolation)

Example 2: Draw a box passing through the points

```
P1 = [50,0,0]';
P2 = [50,30,0]';
P3 = [20,30,0]';
P4 = [20,0,0]';


Q1 = MoveTo(P1, P2, 3);
Q2 = MoveTo(P2, P3, 3);
Q3 = MoveTo(P3, P4, 3);
Q4 = MoveTo(P4, P1, 3);

Q = [Q1, Q2, Q3, Q4];

t = [1:length(Q)]' * 0.01;
clf
plot(t,Q)
```



Resulting Joint Angles:



Joint Angles for drawing a square with MoveTo command

CircleXY()

The equation for a circle is

$$x = x_0 + r \cdot \cos\theta$$

$$y = y_0 + r \cdot \sin\theta$$

where

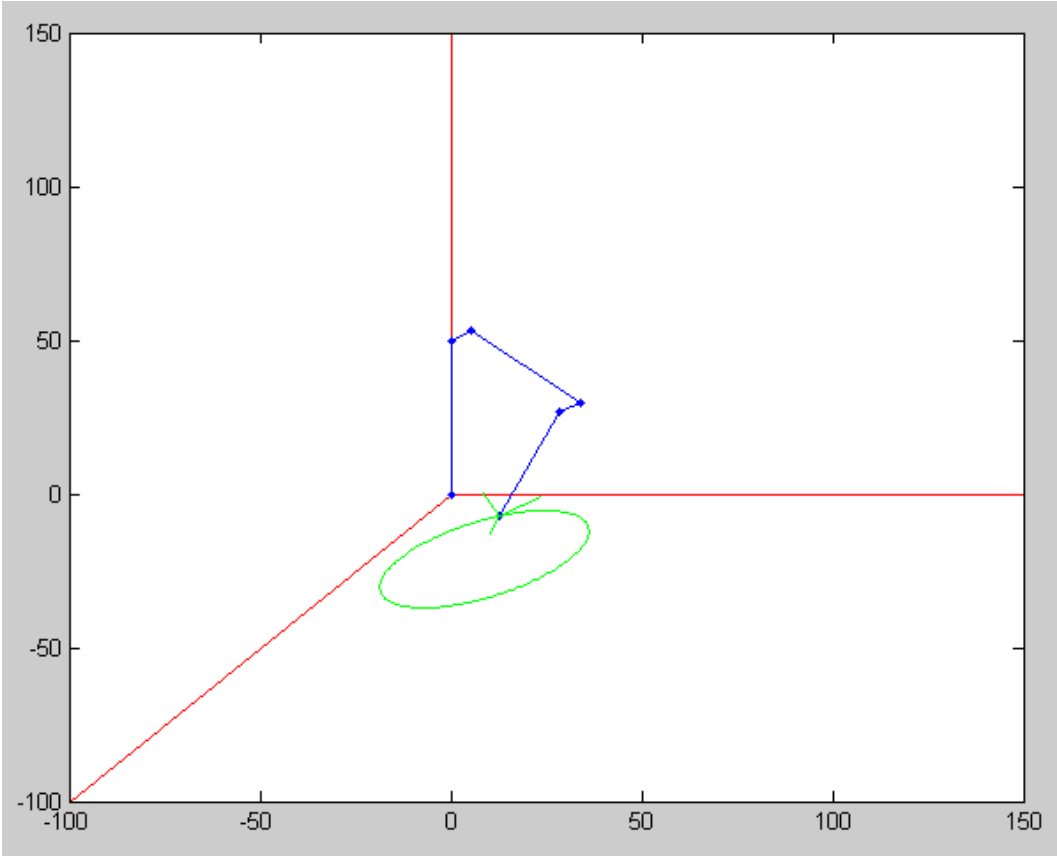
- (x_0, y_0) is the center of the circle
- r is the radius, and
- $0 < \theta < 2\pi$

Code:

```
function [Q] = CircleXY(P0, P1, T)

% Draw a cirle in the XY plane with center at P1
% starting at point P0
% in T seconds
```

Draw a circle, starting at point [10,20,30]', centered at [30,30,30]', in 3 seconds

```
Q = CircleXY([10,20,0]',  [30,30,0]',  3);
```



ArcXY(P1, P2, P3, T)

Draw an arc in the XY plane, starting from point P1, passing through point P2, ending at point P3, in T seconds.

The mathematics for this is as follows. Any point on a circle is defined as

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

Plugging in points for P1, P2, P3 results in three equations for three unknowns (x0, y0, r).

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = r^2$$

$$(x_2 - x_0)^2 + (y_2 - y_0)^2 = r^2$$

$$(x_3 - x_0)^2 + (y_3 - y_0)^2 = r^2$$

Set equation 1 equal to the second and third results in two equations for two unknowns (x0, y0)

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = r^2 = (x_2 - x_0)^2 + (y_2 - y_0)^2$$
$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = r^2 = (x_3 - x_0)^2 + (y_3 - y_0)^2$$

Multiply these out

$$\left(x_1^2 - 2x_1x_0 + x_0^2\right) + \left(y_1^2 - 2y_1y_0 + y_0^2\right) = \left(x_2^2 - 2x_2x_0 + x_0^2\right) + \left(y_2^2 - 2y_2y_0 + y_0^2\right)$$
$$\left(x_1^2 - 2x_1x_0 + x_0^2\right) + \left(y_1^2 - 2y_1y_0 + y_0^2\right) = \left(x_3^2 - 2x_3x_0 + x_0^2\right) + \left(y_3^2 - 2y_3y_0 + y_0^2\right)$$

Simplify

$$\left(x_1^2 - 2x_1x_0\right) + \left(y_1^2 - 2y_1y_0\right) = \left(x_2^2 - 2x_2x_0\right) + \left(y_2^2 - 2y_2y_0\right)$$
$$\left(x_1^2 - 2x_1x_0\right) + \left(y_1^2 - 2y_1y_0\right) = \left(x_3^2 - 2x_3x_0\right) + \left(y_3^2 - 2y_3y_0\right)$$

Group terms

$$\begin{bmatrix} (2x_2 - 2x_1) & (2y_2 - 2y_1) \\ (2x_3 - 2x_1) & (2y_3 - 2y_1) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} \left(x_2^2 + y_2^2\right) - \left(x_1^2 + y_1^2\right) \\ \left(x_3^2 + y_3^2\right) - \left(x_1^2 + y_1^2\right) \end{bmatrix}$$

Solve for the center of the circle (x0, y0). The radius is from any of these equations:

$$r^2 = (x_1 - x_0)^2 + (y_1 - y_0)^2$$

The initial and final angle are from

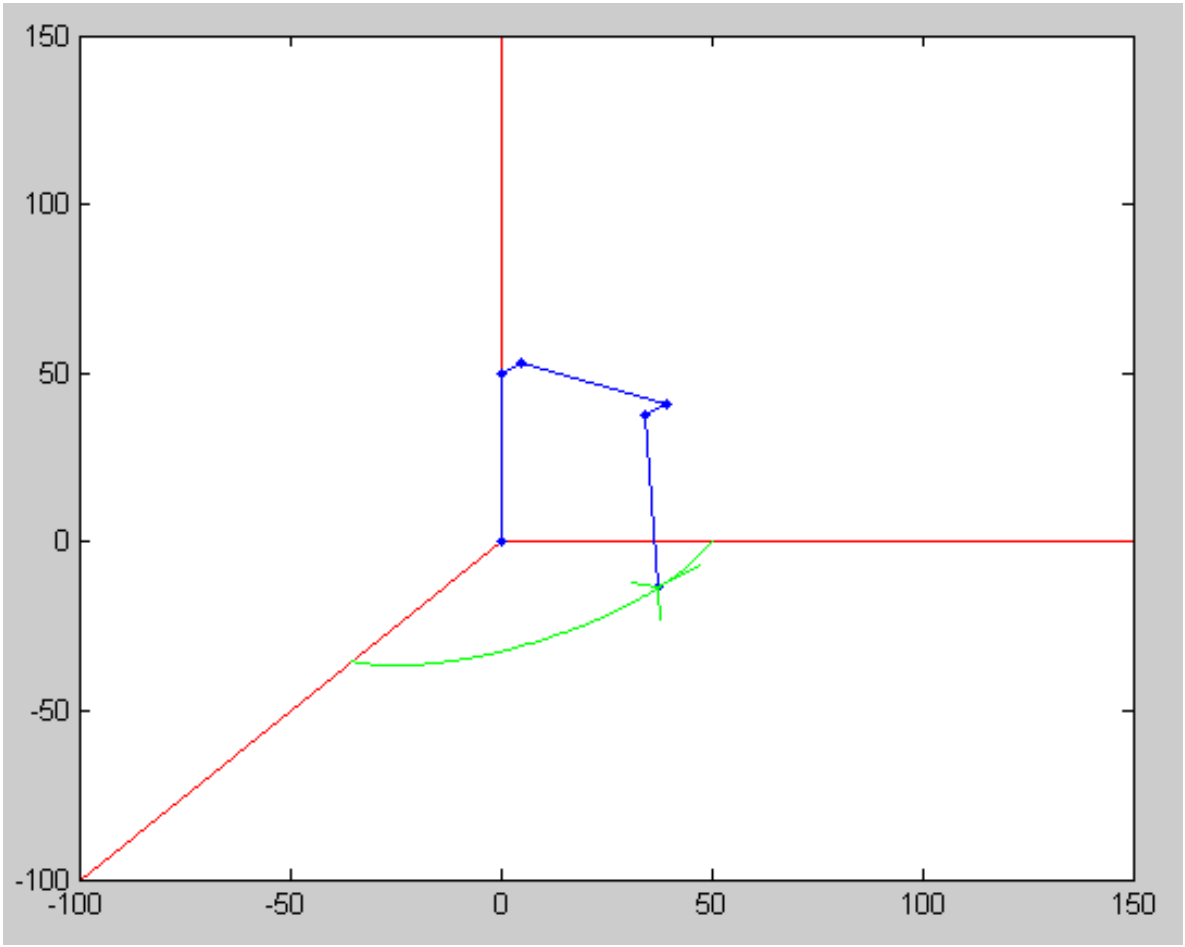
$$\theta_1 = \arctan\left(\frac{y_1 - y_0}{x_1 - x_0}\right)$$

$$\theta_3 = \arctan\left(\frac{y_3 - y_0}{x_3 - x_0}\right)$$

The arc goes from these two angles, with 2π added or subtracted to keep from going the long way around the circle if the angles go past zero (i.e. an arc from 350 degrees to 10 degrees should really go from -10 to +10 degrees.).

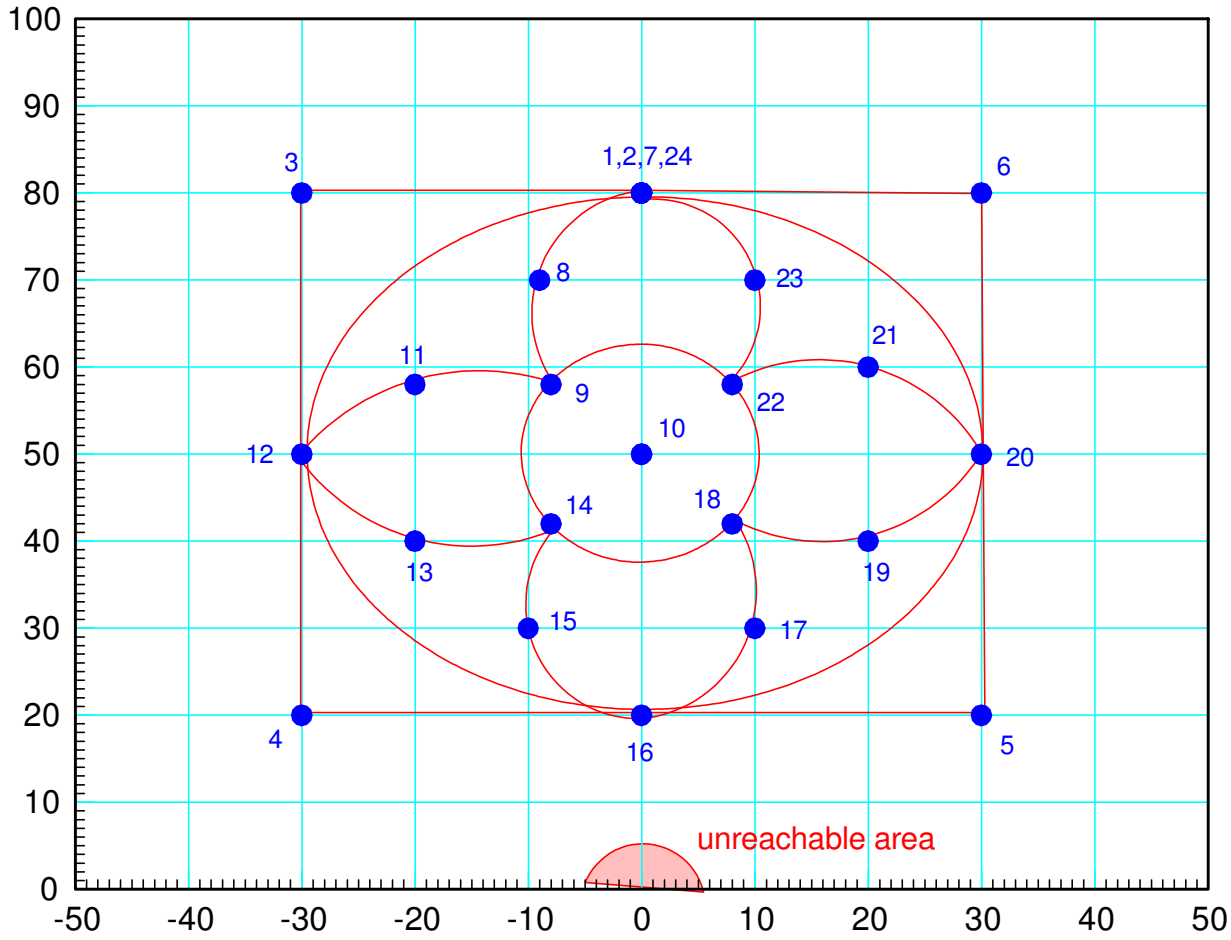
Sample Code: Draw a

```
Q = ArcXY([50,0,0]',  [40,40,0]',  [0,50,0]', 3);
```

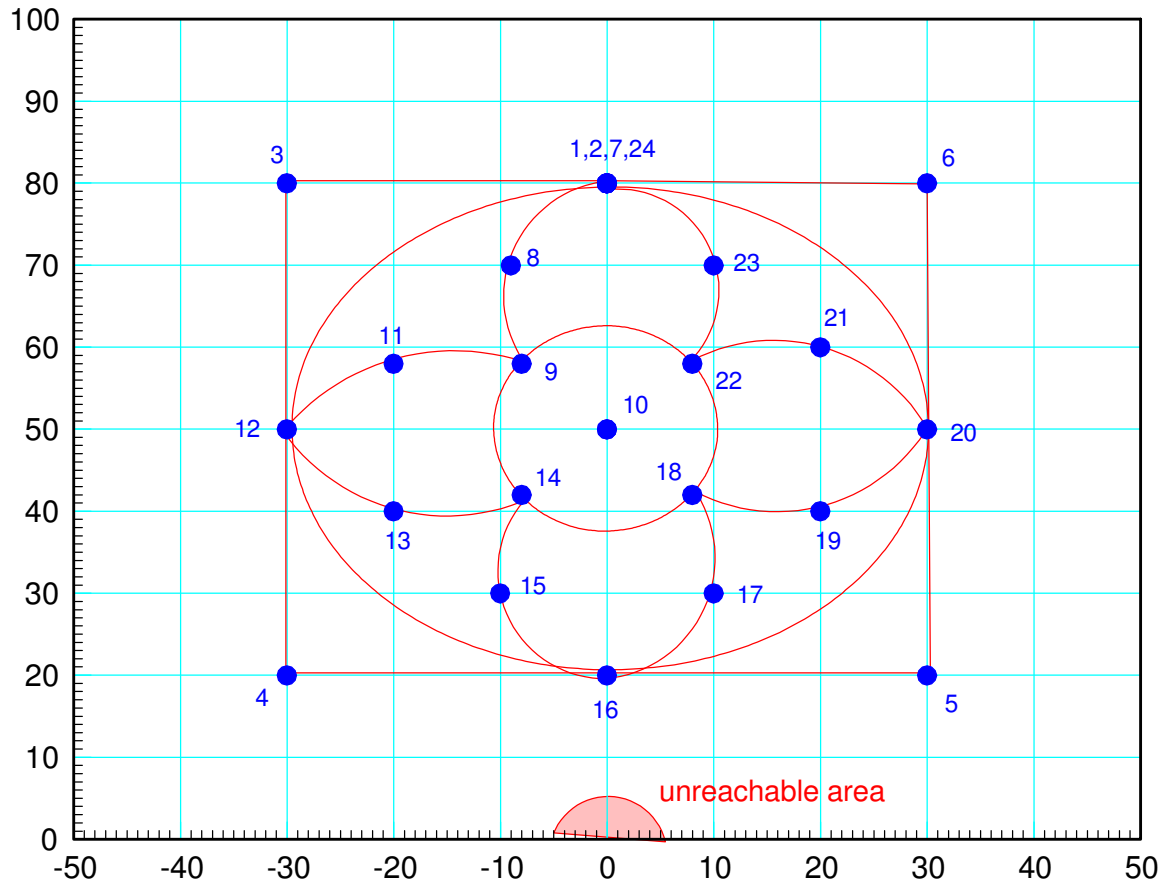


Program: Draw a Dogwood Symbol:

```
P1 = [0; 80; 20];
P2 = [0; 80; 0];
P3 = [-30; 80; 0];
P4 = [-30; 20; 0];
P5 = [30; 20; 0];
P6 = [30; 80; 0];
P7 = [0; 80; 0];
P8 = [-9; 70; 0];
P9 = [-8; 58; 0];
P10 = [0; 50; 0];
P11 = [-20; 58; 0];
P12 = [-30; 50; 0];
P13 = [-20; 40; 0];
P14 = [ -8; 42; 0];
P15 = [-10; 30; 0];
P16 = [  0; 20; 0];
P17 = [ 10; 30; 0];
P18 = [  8; 42; 0];
P19 = [ 20; 40; 0];
P20 = [ 30; 50; 0];
P21 = [ 20; 60; 0];
P22 = [  8; 58; 0];
P23 = [ 10; 70; 0];
P24 = [  0; 80; 0];
P25 = [0; 80; 20];
```

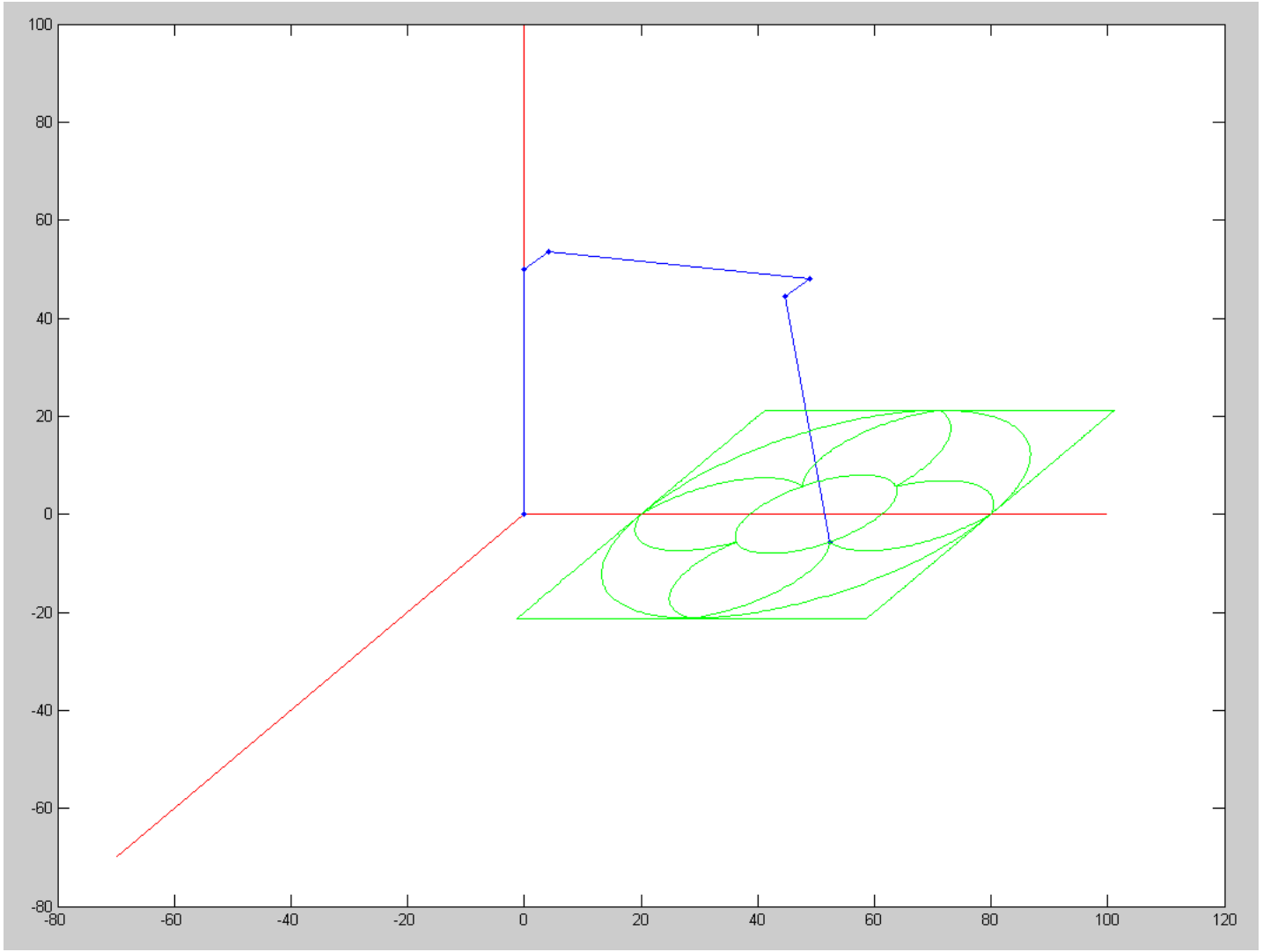


```
% pen down
Q1 = MoveTo(P1, P2, 1);
% draw a box
Q2 = MoveTo(P2, P3, 2);
Q3 = MoveTo(P3, P4, 2);
Q4 = MoveTo(P4, P5, 2);
Q5 = MoveTo(P5, P6, 2);
Q6 = MoveTo(P6, P7, 2);
% draw a circle
Q7 = CircleXY(P7, P10, 4);
% draw arcs
Q8 = ArcXY(P7, P8, P9, 1);
% draw a circle
Q9 = CircleXY(P9, P10, 2);
% draw arcs
Q10 = ArcXY(P9, P11, P12, 1);
Q11 = ArcXY(P12, P13, P14, 1);
Q12 = ArcXY(P14, P15, P16, 1);
Q13 = ArcXY(P16, P17, P18, 1);
Q14 = ArcXY(P18, P19, P20, 1);
Q15 = ArcXY(P20, P21, P22, 1);
Q16 = ArcXY(P22, P23, P24, 1);
% pen up
Q17 = MoveTo(P24, P1, 1);
```

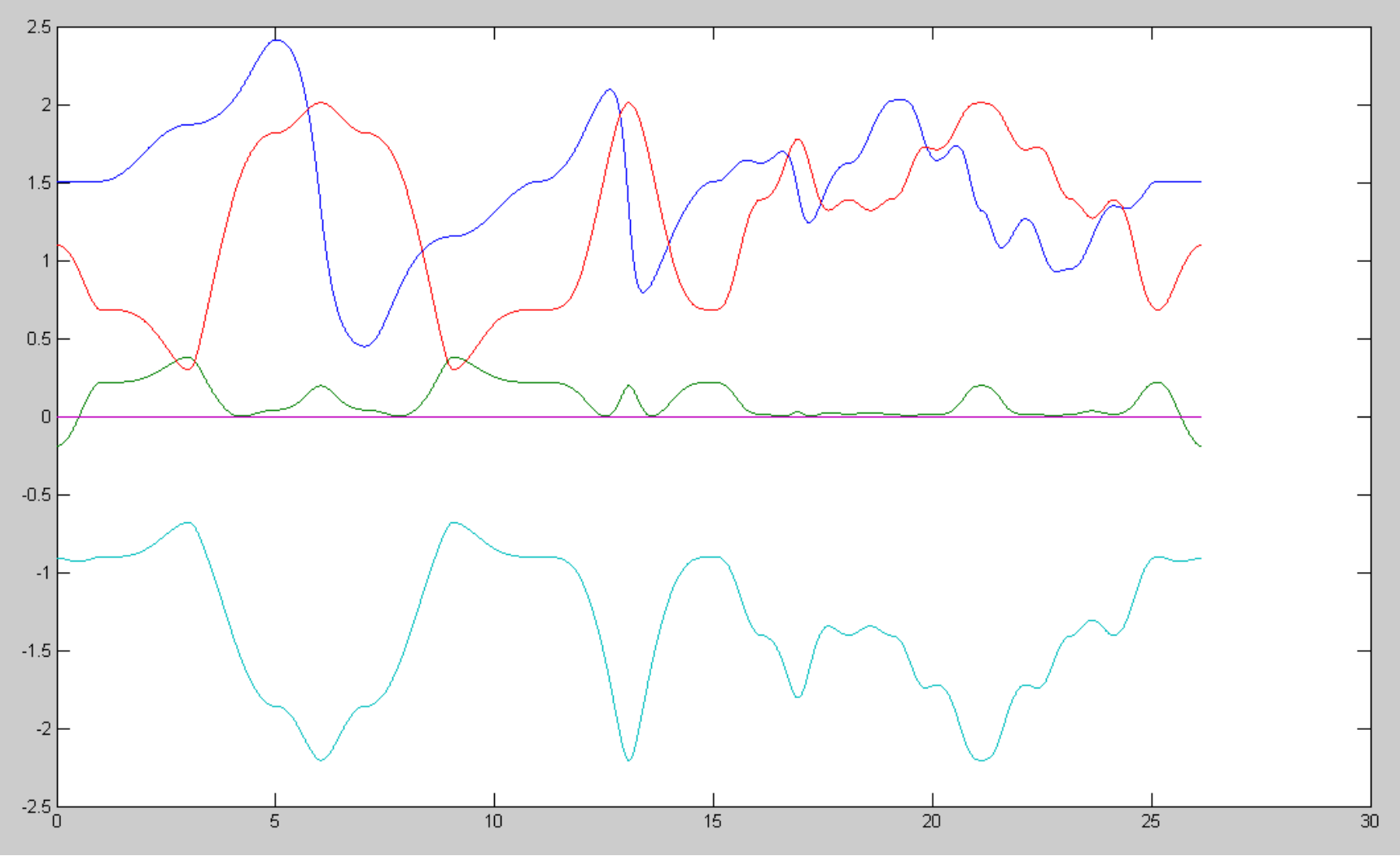



```
Q = [Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17];
```

Resulting tip position:



Resulting joint angles (the program)



Homework #8

- (2-day assignment)
- Define a shape to map out (your pick)
- Write a program to implement this shape
- Plot the resulting joint angles