

ECE 111 - Homework #3

Week #3: Trigonometry

Polar to Rectangular Conversions

1) Determine the final position of A: (x,y)

$$A = (8\angle 46^0) + (6\angle 125^0) + (7\angle 66^0)$$

in Matlab

```
>> x1 = 8 * cosd(46);  
>> y1 = 8 * sind(46);  
  
>> x2 = 6 * cosd(125);  
>> y2 = 6 * sind(125);  
  
>> x3 = 7 * cosd(66);  
>> y3 = 7 * sind(66);  
  
>> Ax = x1 + x2 + x3  
  
Ax = 4.9630  
  
>> Ay = y1 + y2 + y3  
  
Ay = 17.0644  
  
>>
```

2) Determine final position of B: (x,y)

$$B = (13\angle 138^0) + (9\angle -155^0) + (7\angle 21^0)$$

In Matlab

```
>> x1 = 13 * cosd(138);  
>> y1 = 13 * sind(138);  
  
>> x2 = 9*cosd(-155);  
>> y2 = 9 * sind(-155);  
  
>> x3 = 7 * cosd(21);  
>> y3 = 7 * sind(21);  
  
>> Bx = x1 + x2 + x3  
  
Bx = -11.2826  
  
>> By = y1 + y2 + y3  
  
By = 7.4037
```

3) Where is A relative to B (i.e. what is $C = B - A$?)

- In (x,y) coordinates
- In polar coordinates

In Matlab

```
>> Cx = Bx - Ax
```

```
Cx = -16.2456
```

```
>> Cy = By - Ay
```

```
Cy = -9.6607
```

In polar coordinates using natural units (radians)

```
>> Cr = sqrt(Cx^2 + Cy^2)
```

```
Cr = 18.9010
```

```
>> Cq = atan2(Cy, Cx)
```

```
Cq = -2.6051
```

$$C = 18.9010 \angle -2.60 \text{ radians}$$

If you prefer degrees

```
>> Cq = atan2(Cy, Cx) * 180/pi
```

```
Cq = -149.2613
```

$$C = 18.9010 \angle -149.26^\circ$$

Plotting Polar Functions

4) Plot the following functions in Matlab for $-2\pi < \theta < 2\pi$

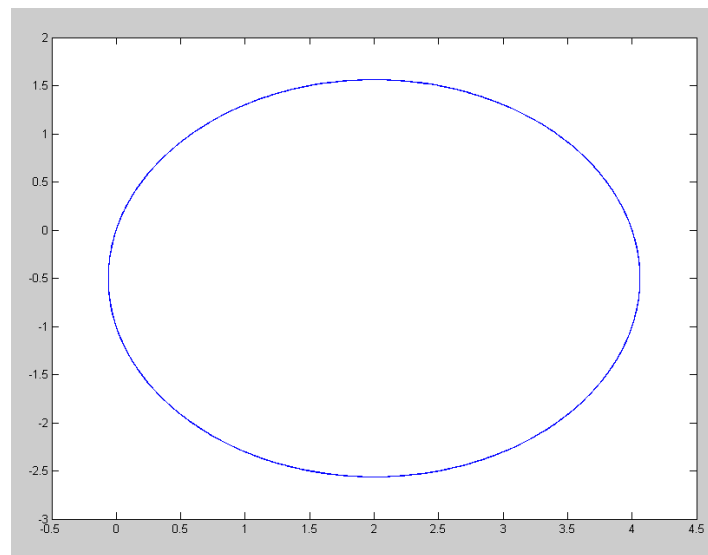
- Note: `plot()` plots in cartesian coordinates. Each function needs to be converted from polar to rectangular.

a) $r = 4 \cos(\theta) - \sin(\theta)$

In Matlab

```
>> q = [-2*pi : 0.01: 2*pi]';  
>> r = 4 * cos(q) - sin(q);  
  
>> x = r .* cos(q);  
>> y = r .* sin(q);  
  
>> plot(x, y)
```

Not too surprising: you get a circle. Trig functions are all about circles.

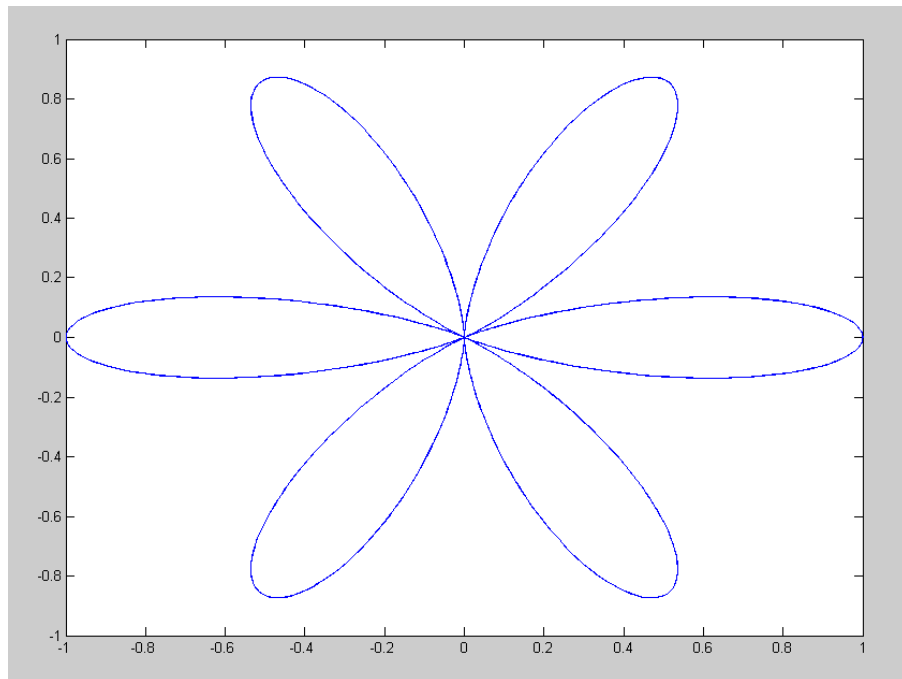


b) $r = (\cos(3\theta))^2$

Matlab Code:

```
>> q = [-2*pi : 0.01: 2*pi]';  
>> r = ( cos(3*q) ) .^ 2;  
>> x = r .* cos(q);  
>> y = r .* sin(q);  
>> plot(x, y)  
>>
```

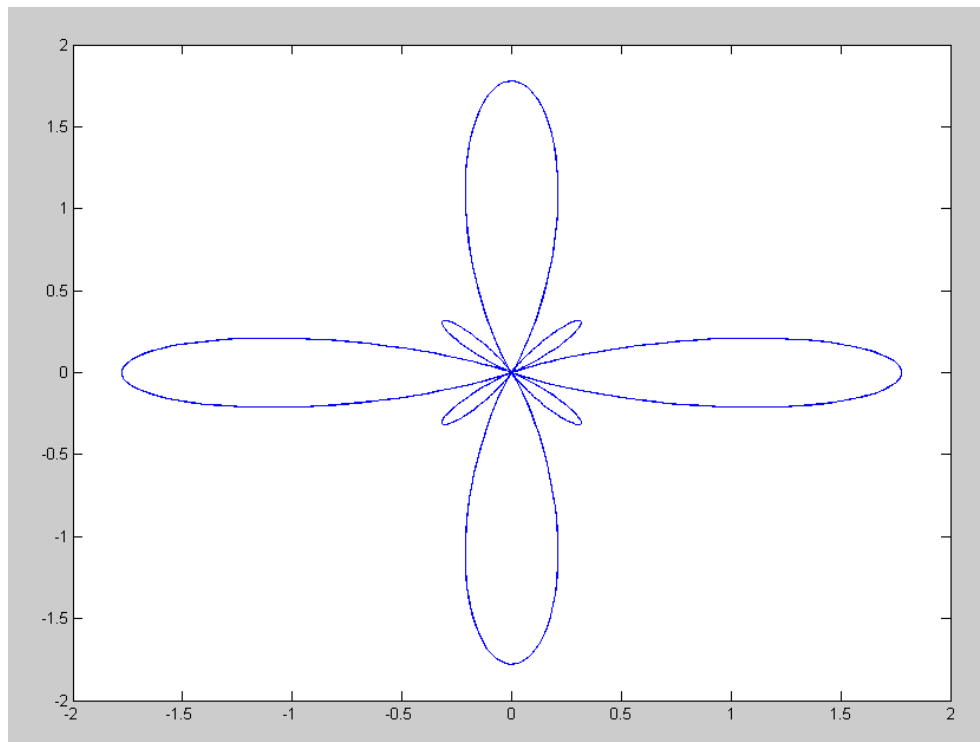
note: The dot-notation is needed to tell Matlab to do 1200 separate operations rather than treat the variables as matrices



c) $r = \left(\frac{1}{3} + \cos(4\theta) \right)^2$

In Matlab

```
>> q = [-2*pi : 0.01: 2*pi]';  
>> r = ( 1/3 + cos(4*q) ) .^ 2;  
>> x = r .* cos(q);  
>> y = r .* sin(q);  
>> plot(x, y)  
>>
```



Robot Tip Position (Forward Kinematics)

A 2D robot has three arms with lengths of {55, 45, 35} cm. The final tip position is

$$x_1 = 55 \cos(\theta_1)$$

$$y_1 = 55 \sin(\theta_1)$$

$$x_2 = x_1 + 45 \cos(\theta_1 + \theta_2)$$

$$y_2 = y_1 + 45 \sin(\theta_1 + \theta_2)$$

$$x_3 = x_2 + 35 \cos(\theta_1 + \theta_2 + \theta_3)$$

$$y_3 = y_2 + 35 \sin(\theta_1 + \theta_2 + \theta_3)$$

5) Plot the tip position (x3, y3) for

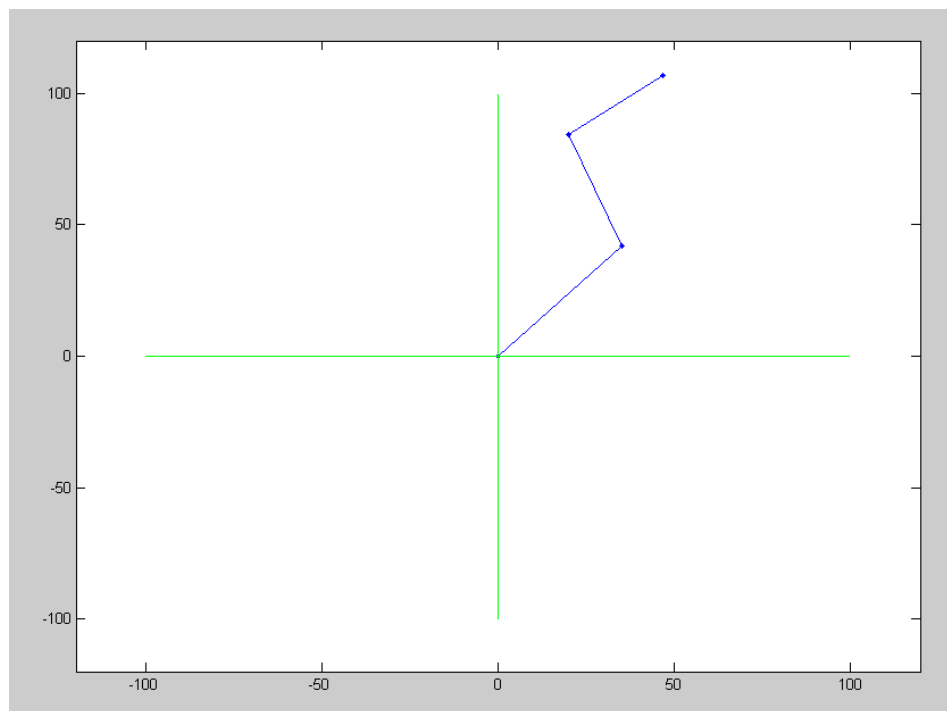
$$\theta_1 = 50^\circ \quad \theta_2 = 60^\circ \quad \theta_3 = -70^\circ$$

Using a Matlab function (next page)

```
>> [x3, y3] = RRR(50, 60, -70)
```

```
x3 = 46.7740
```

```
y3 = 106.9162
```



Planning ahead, create a Matlab function where

- You pass the angles,
- It draws the robot (just because), and
- It returns the tip position

```
function [x,y] = RRR(q1,q2,q3)
    x0 = 0;
    y0 = 0;
    % calculate joint positions
    x1 = 55 * cosd(q1);
    y1 = 55 * sind(q1);

    x2 = x1 + 45 * cosd(q1+q2);
    y2 = y1 + 45 * sind(q1+q2);

    x3 = x2 + 35 * cosd(q1+q2+q3);
    y3 = y2 + 35 * sind(q1+q2+q3);

    % plot the robot's position
    plot([x0,x1,x2,x3],[y0,y1,y2,y3],'b.-');
    xlim([-120,120]);
    ylim([-120,120]);
    pause(0.1);

    % return the tip position
    x = x3;
    y = y3;
end
```

6) Plot the tip position (x3, y3) for

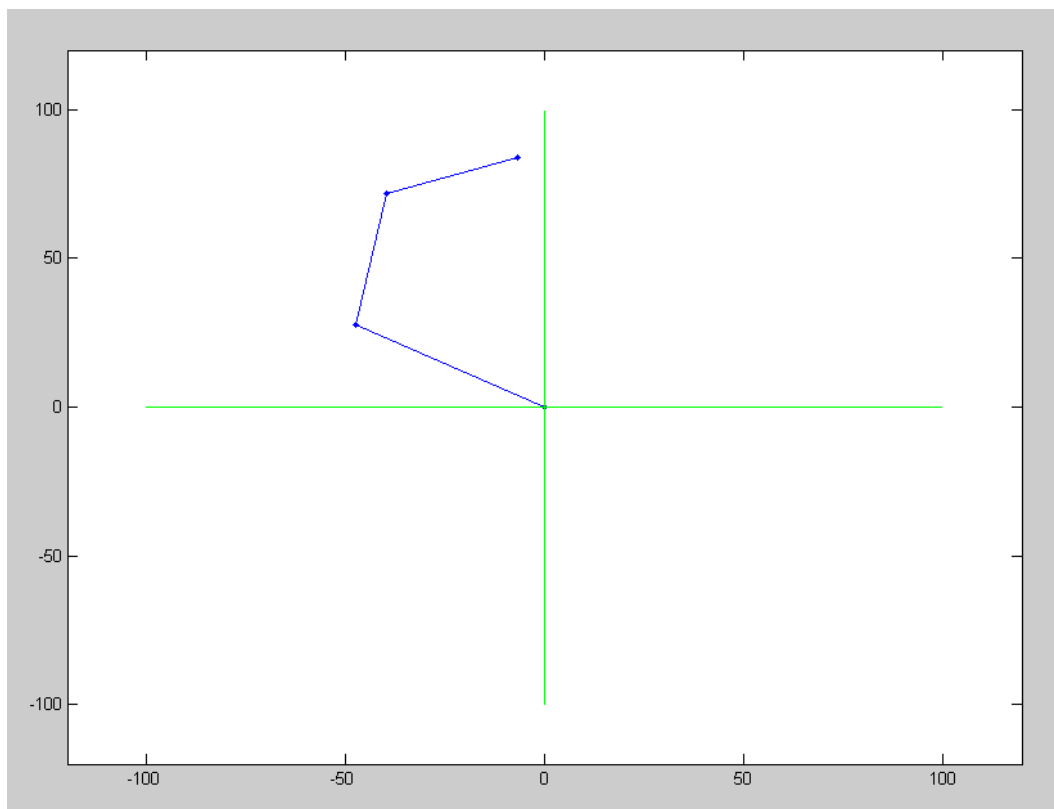
$$\theta_1 = 150^\circ \quad \theta_2 = -70^\circ \quad \theta_3 = -60^\circ$$

In Matlab

```
>> [x3, y3] = RRR(150, -70, -60)
```

```
x3 =    -6.9280
```

```
y3 =    83.7871
```



Robot Tip Position (Inverse Kinematics & fminsearch())

7) Write a Matlab function which

- Is passed the angles ($\theta_1, \theta_2, \theta_3$),
- Computes the tip position, and
- Returns the distance from the tip position to ($x_3 = 70\text{cm}$, $y_3 = 70\text{cm}$)

In Matlab

```
function [e] = Prob7(z)
    q1 = z(1);
    q2 = z(2);
    q3 = z(3);

    [x3,y3] = RRR(q1,q2,q3);

    dx = x3 - 70;
    dy = y3 - 70;

    e = dx^2 + dy^2;
end
```

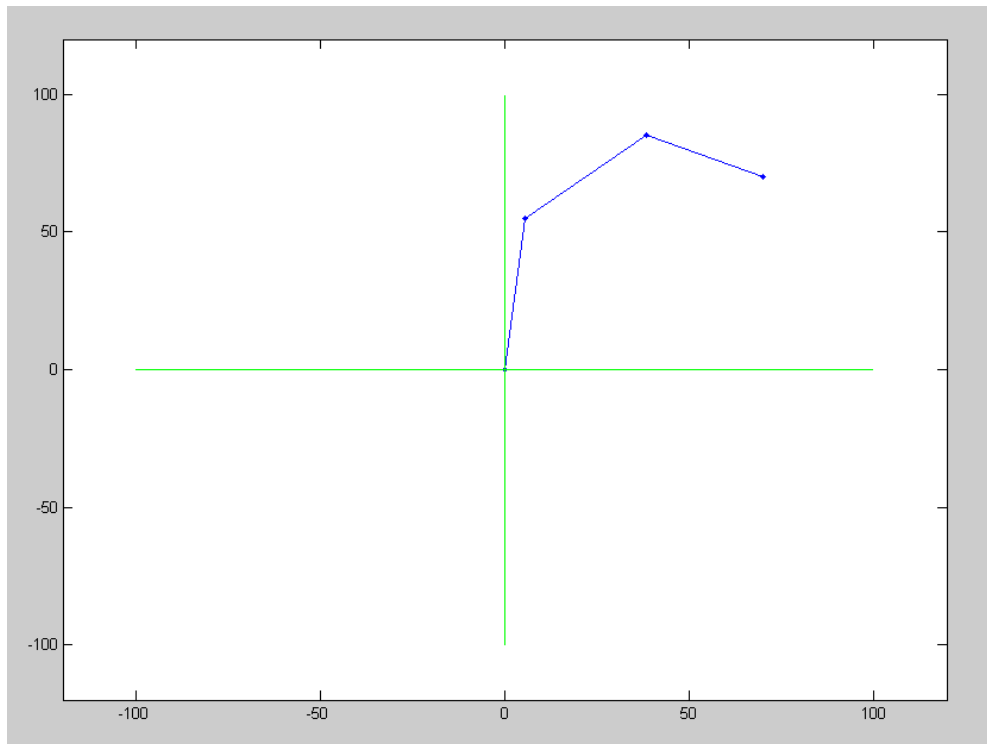
8) Use the `fminsearch()` to determine the joint angles which place the robot at ($x_3 = 70\text{cm}$, $y_3 = 70\text{cm}$)

Use `fminsearch()` to drive the error to zero. The initial guess affects which solution you wind up with

```
>> [Z,e] = fminsearch('Prob7',[150, -70, -60])
```

```
      q1      q2      q3  
Z = 84.2225 -41.3791 -68.8025
```

```
e = 6.2669e-014
```

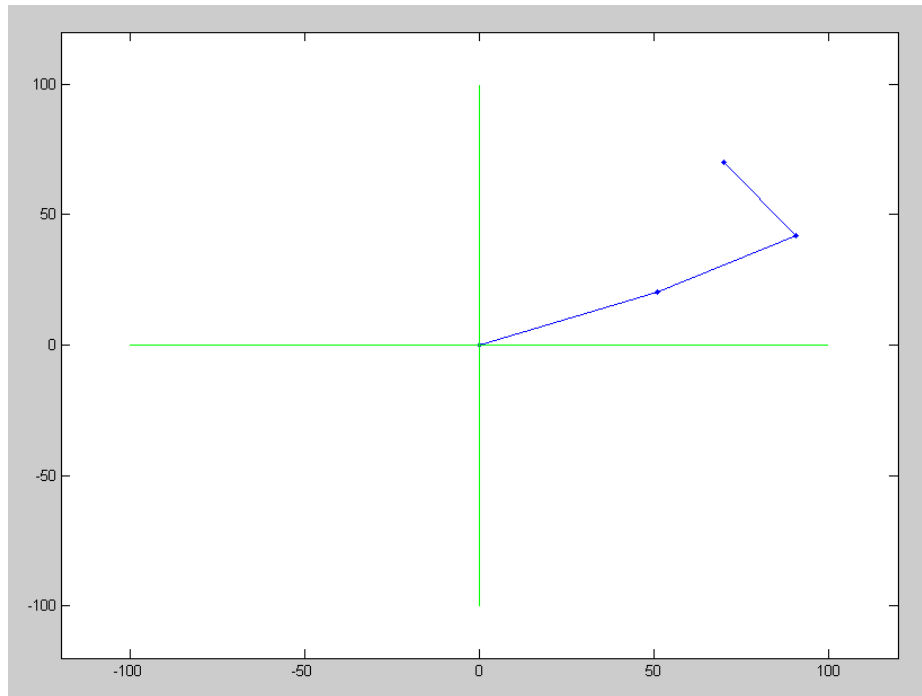


A different initial guess gives a different solution (there are an infinite number of valid solutions)

```
>> [Z,e] = fminsearch('Prob7',[50, 60, 70])
```

```
z =      q1      q2      q3  
    21.8339    6.4355   97.9636
```

```
e = 3.2672e-012
```



Comment:

- *fminsearch()* is a really useful function.
- You can use it to solve all sorts of problems. The trick is to set up the problem so that the solution is a minimum