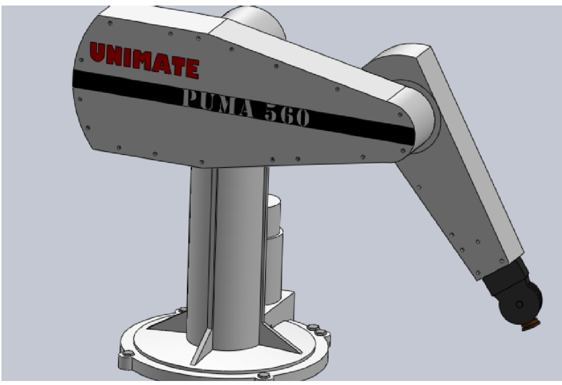
Inverse Kinematics of a Puma Robot

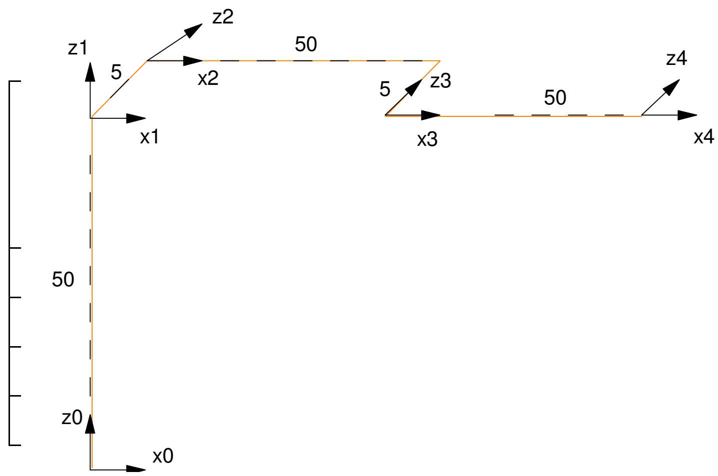
Forward Kinematics is computing the tip position given the angles.

Inverse Kinematics is computing the joint angles given the tip position.

Inverse Kinematics for a Puma robot:

Consider first the RRR robot with no net offset for the tip (d2 + d3 = 0)



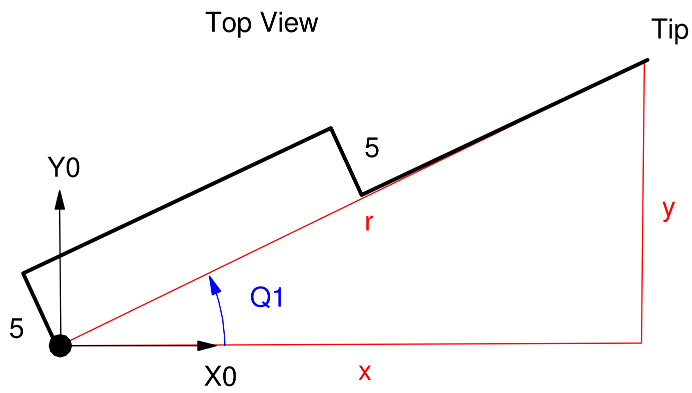


RRR Robot (similar to a Puma Robot)

Link i	α_{i-1} The angle between the Zi-1 and Zi axis (twist)	a_{i-1} The distance from Zi-1 to Zi measured along the Xi-1 axis	d_i The distance from Xi-1 to Xi measured along the Zi axis	Q_i The angle between Xi-1 and Xi measured about the Zi axis
1	0	0	50	Q1
2	-90 deg	0	5	Q2
3	0	50	-5	Q3
4 (tip)	0	50	0	0

The trick with inverse kinematics is to determine the joint angles given the tip position.

Since d2 + d3 = 0, the net effect is the tip is inline with the base of the robot. A top view (seen looking down the Z0 axis) is:



Top View of the RRR robot. Note that the two offsets cancel resulting in Q1 point to the tip.

The first joint angle is then

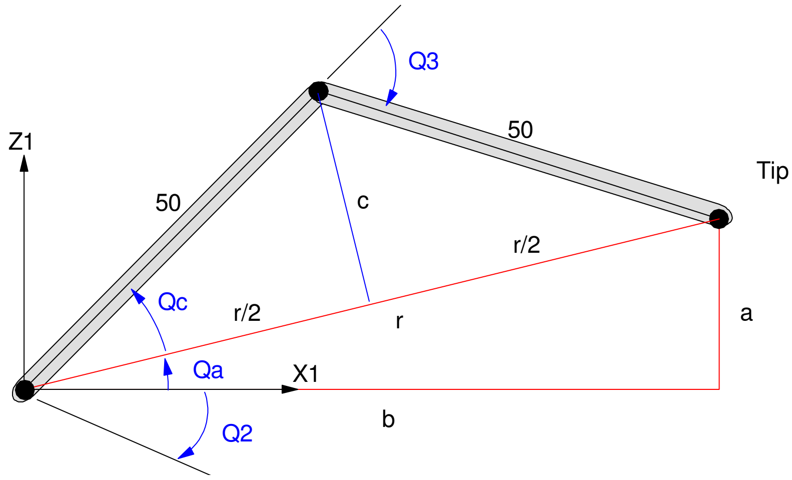
$$\theta_1 = \arctan\left(\frac{y_{tip}}{x_{tip}}\right)$$

In Matlab, this is the function *atan2*, which returns the angle from 0 to 360 degrees

```
Q1 = atan2(Ytip, Xtip)
```

Next, looking at the side of the robot, try to determine the joint angles Q2 and Q3. First, assume that Q1 = 0. With this assumption, the side view looks at the YZ plane.

Next, assume that the origin is at axis (X1, Y1, Z1). The tip position relative to the (X1, Y1, Z1) axis is as follows:



Side view with Q1 = 0 degrees, relative to the shoulder joint (reference frame 1):

Here

$$a = z_{tip}$$

$$b = x_{tip}$$

$$r = \sqrt{b^2 + z_{tip}^2}$$

$$c = \sqrt{50^2 - \left(\frac{r}{2}\right)^2}$$

and

$$\theta_a = \arctan\left(\frac{a}{b}\right)$$

$$\theta_c = \arctan\left(\frac{c}{r/2}\right)$$

The shoulder joint angle is then

$$\theta_2 = -(\theta_a + \theta_c)$$

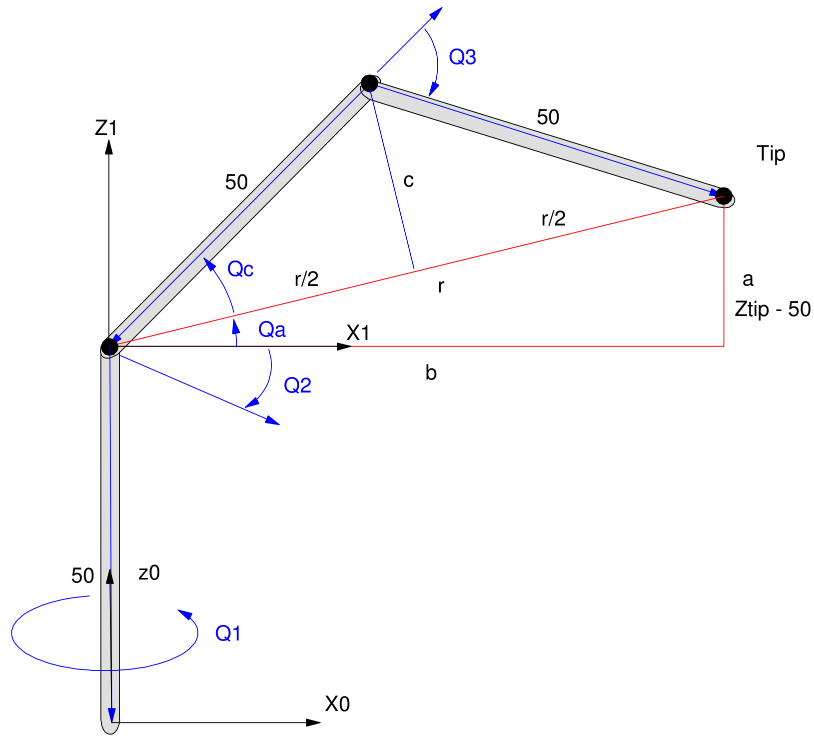
and the elbow joint angle is

$$\theta_3 = 2\theta_c$$

If θ_1 is not zero degrees, then replace 'b' with the distance to the origin, r

$$b = \sqrt{x_{tip}^2 + y_{tip}^2}$$

For the RRR robot, the shoulder joint is 50cm above the base, so replace 'z' with 'z-50'



Sice view of the Puma robot. Note that the shoulder joint is offset by 50cm

The net result is the following Matlab code is on the following page. You can check this by

- Input a point for the desired tip position
- Find the angles which get you to this point (InversePuma)
- Find the resulting point these angles give you (Puma)

The two should match. For example, to move to the the point (30, 50, 70), the angles are

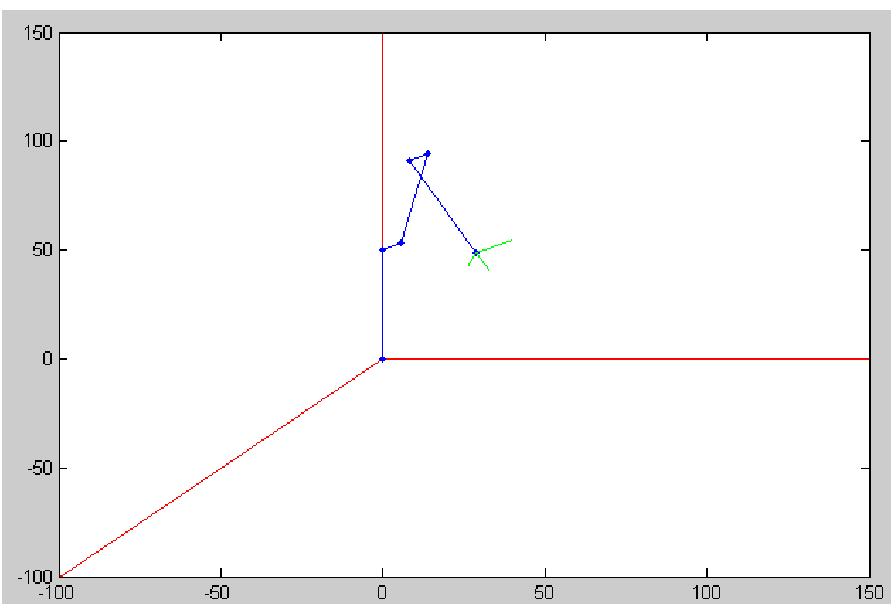
```
>> TIP = [30,50,70]'
>> Q = InversePuma(TIP)

    1.0304
   -1.2370
    1.8132
```

These angles put the robot at the following point:

```
>> Puma(Q,TIP)

   30.0000
   50.0000
   70.0000
    1.0000
```

Check - the resulting position is where it's supposed to be.

```
function [Q] = InversePuma(TIP)

   Xtip = TIP(1);
   Ytip = TIP(2);
   Ztip = TIP(3);

   w = sqrt(Xtip^2 + Ytip^2)
   b = sqrt(w^2 - 5^2)
   Qd = atan2(5, b)
   Qtip = atan2(Ytip, Xtip)

   q1 = Qtip - Qd;

   a = Ztip - 50;
   r = sqrt(b^2 + a^2);
   c = sqrt(50^2 - (r/2)^2);

   qa = atan2(a, b);
   qc = atan(c / (r/2) );

   q2 = - ( qa + qc );
   q3 = 2*qc;

   Q = [q1; q2; q3];

   end
```



Verification of the InversePuma routine: The joint angles put you at (30, 50, 70) as desired

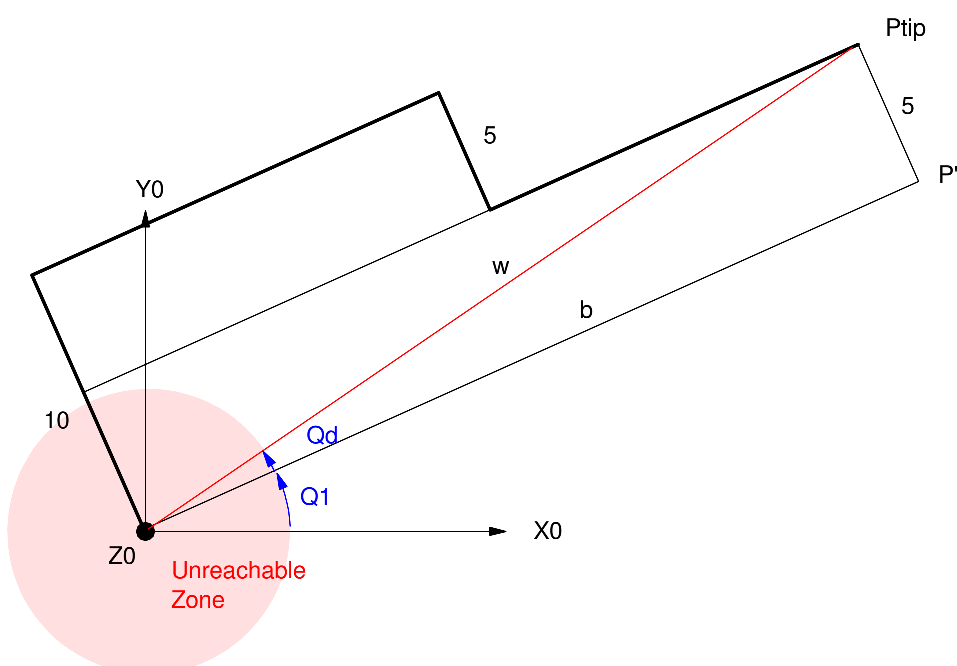
Case 2: d2 + d3 ≠ 0

For the Pume robot, d2 and d3 are *not* the same.

Link i	α_{i-1} The angle between the Zi-1 and Zi axis (twist)	a_{i-1} The distance from Zi-1 to Zi measured along the Xi-1 axis	d_i The distance from Xi-1 to Xi measured along the Zi axis	Q_i The angle between Xi-1 and Xi measured about the Zi axis
1	0	0	50	Q1
2	-90 deg	0	10	Q2
3	0	50	-5	Q3
4 (tip)	0	50	0	0

This means, looking down the Z0 axis, there is a 5cm offset of the tip from the reference frame. The result is

- There is a cyllinder about the Z0 axis with a radius of 5cm where the robot cannot reach
- The equations for the inverse kinematics get a bit more complicated.



Top View of the PUMA Robot. The shoulder and elbow offsets do not cancel, resulting in a 5cm offset for the tip

First, determine the joint angle, Q1

$$w = \sqrt{x_{tip}^2 + y_{tip}^2}$$

$$b = \sqrt{w^2 - 5^2}$$

$$\theta_{tip} = \arctan\left(\frac{y_{tip}}{x_{tip}}\right)$$

$$\theta_d = \arctan\left(\frac{5}{b}\right)$$

$$\theta_1 = \theta_{tip} - \theta_d$$

From this point on, the previous equations all apply, with the note that 'b' is the value computed here rather than the distance to the tip (called 'w' here)

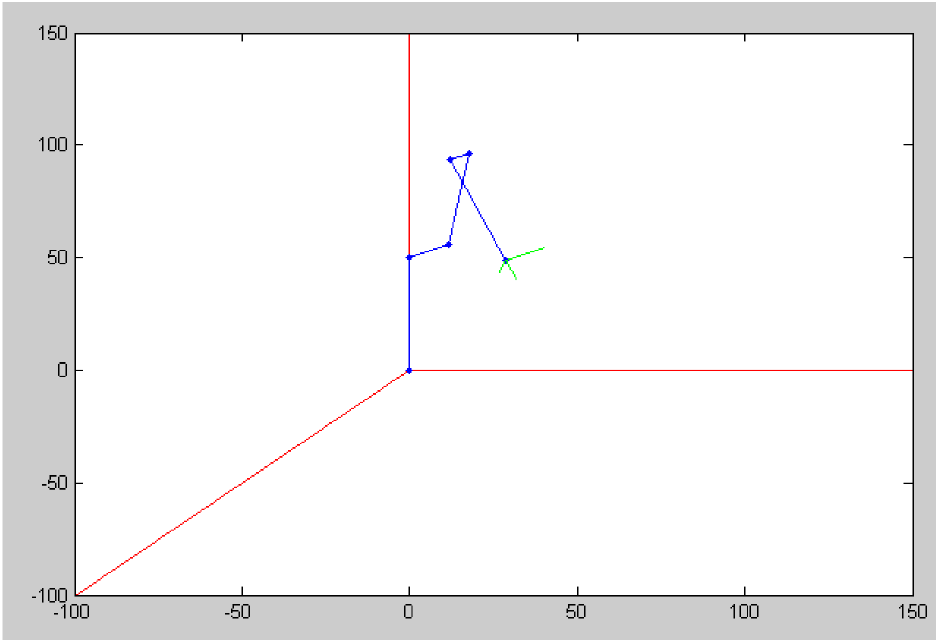
The resulting Matlab code is then

```
function [Q] = InversePuma(TIP)

   Xtip = TIP(1);
   Ytip = TIP(2);
   Ztip = TIP(3);

   w = sqrt(Xtip^2 + Ytip^2);
   b = sqrt(w^2 - 5^2);
   Qd = atan2(5, b);
   Qtip = atan2(Ytip, Xtip);

   q1 = Qtip - Qd;

   a = Ztip - 50;
   r = sqrt(b^2 + a^2);
   c = sqrt(50^2 - (r/2)^2);

   qa = atan2(a, b);
   qc = atan(c / (r/2) );

   q2 = - ( qa + qc );
   q3 = 2*qc;

   Q = [q1; q2; q3];

   end
```

Checking the code by finding the angles which place you at point (30, 50, 70) cm then finding where these angles place you:

```
TIP = [30,50,70]';
Q = InversePuma(TIP)

    0.9445
   -1.2407
    1.8183

Puma(Q, TIP)

    30
    50
    70
     1
```

Check: The joint angles place you at the correct position.



Verification that the InversePuma routine works: The joint angles place you at (30, 50, 70)