ME 598
Introduction to Robotics
Fall 2022

Lab 1: Robot Arm Kinematics

Group 8
10/17/22

Undergraduate:
"I pledge my honor that I have abided by the Stevens Honor System"

Graduate:
"I pledge that I have abided by the Graduate Student Code of Academic Integrity."

The report has been prepared by:
1. Rakesh Gummapu
2. Kevin Perez
3. Shiv Bhatt
4. Shravil Sing

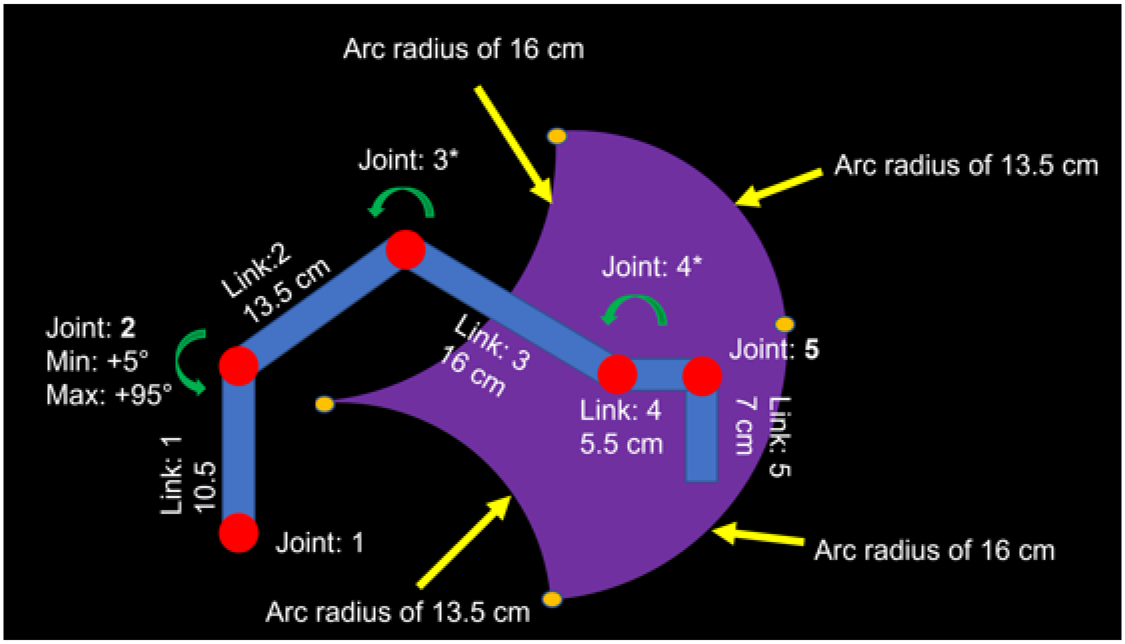
Abstract
The lab_1 deals with the concepts of Forward Kinematics and Inverse Kinematics of the Robot arm (DoBot). In this lab work, as a group, Kevin completed the robot's motion profile and reachable workspace, Sharvil did the Forward Kinematics, Shiv has contributed for the Inverse Kinematics and Rakesh has done the Forward and Inverse Kinematics validation, Matlab code for the Lab and made the report.

Introduction
This Lab work has enhanced our understanding on the fundamentals of how the Robot arm works in a 3D workspace and the concepts of DH table, Forward Kinematics and Inverse Kinematics are well understood from this Lab.The reachable workspace and the range of the robot can be found based on the mechanical constraints of the dobot robot arm. From this lab experiment we can see if the robot is working in the given range or is it going out of the workspace depending on the angles and end effector positions.

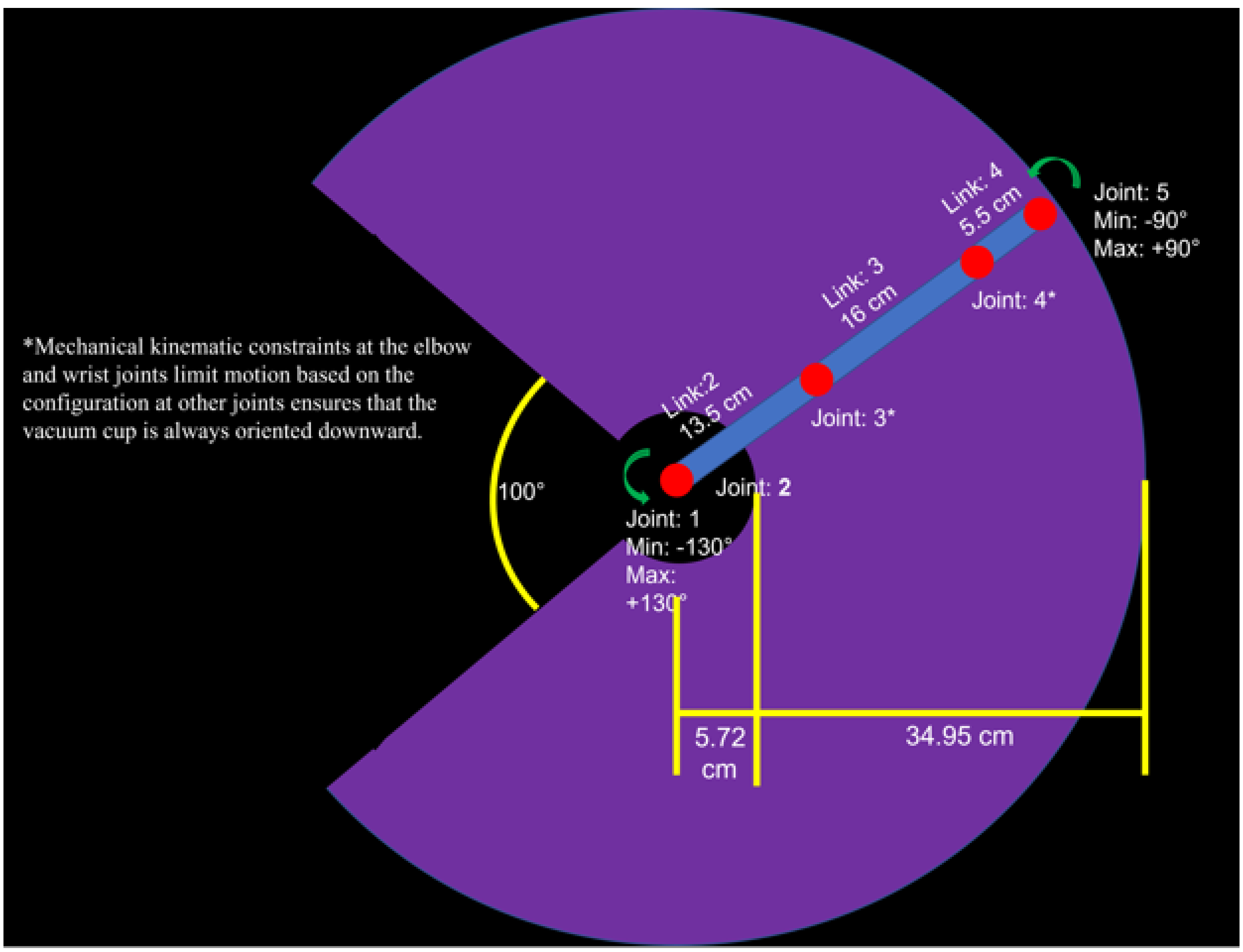
Experimental Procedure
Part-A:
Given an articulated manipulator (RRRRR). The dobot contains 5 revolute joints and 5 links.
The link lengths and the mechanical kinematic constraints are given in the below table.

Link	Length(cm)
1	10.5
2	13.5
3	16
4	5.5
5	7

Joint	Min	Max
1(base)	-135	135
2(shoulder)	5	95
3(elbow)	*	*
4(wrist pitch)	-5	85
5(vacuum twist)	-90	90



Side View



Top view

- Rmax = 34.95cm when theta2 = 5º, theta3 = 30º, theeta4 = 25º
- The minimal degree of joint 3 theoretically varies from -90º to -160º when joint 2 moves from 5º to 75º, the minimal degree of joint 3 rotation is -160º once joint 2 rotates above 75º. The maximum angle of joint 3 decreases when joint 2 rotates more than 35º.
- The above figure top view shows that the dobot base joint rotates from -135º to 135º. Measuring the maximum radius as 34.95cm when setting theta2 = 5º, theta3 = 30º, theta5 = 25º.

Part-B:

- **Forward Kinematics**

The table of DH parameters.

θ	**α (radians)**	**a (cm)**	**d (cm)**
θ_1	$\pi/2$	0	10.5
θ_2	0	13.5	0
θ_3	0	16	0
θ_4	$\pi/2$	5.5	0
θ_5	0	0	7

From the Dh table we can write the homogeneous transform matrix
as
A =Rot(z,theta) Trans(z,d) Trans(x,alpha) Rot(x,alpha)
We write the transform matrix as
A_i =

$$\begin{bmatrix} C_{\theta i} & -S_{\theta i}\, C_{\alpha i} & S_{\theta i}\, S_{\alpha i} & a_i\, C_{\alpha i} \\ S_{\theta i} & C_{\theta i}\, C_{\alpha i} & -C_{\theta i}\, S_{\alpha i} & a_i\, S_{\alpha i} \\ 0 & S_{\alpha i} & C_{\alpha i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$A_1 =$

$$\begin{bmatrix} \mathrm{Cos}\ (t_1) & 0 & \mathrm{Sin}\ (t_1) & 0 \\ \mathrm{Sin}\ (t_1) & 0 & -\mathrm{Cos}\ (t_1) & 0 \\ 0 & 1 & 0 & 10.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$A_2 =$

$$\begin{bmatrix} \mathrm{Cos}\ (t_2) & -\mathrm{Sin}\ (t_2) & 0 & 13.5\mathrm{Cos}\ (t_2) \\ \mathrm{Sin}\ (t_2) & \mathrm{Cos}\ (t_2) & 0 & 13.5\mathrm{Cos}\ (t_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$A_3 =$

$$\begin{bmatrix} \mathrm{Cos}\ (t_3) & -\mathrm{Sin}\ (t_3) & 0 & 16\mathrm{Cos}\ (t_3) \\ \mathrm{Sin}\ (t_3) & \mathrm{Cos}\ (t_3) & 0 & 16\mathrm{Cos}\ (t_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$A_4 =$

$$\begin{bmatrix} \mathrm{Cos}\ (t_4) & 0 & \mathrm{Sin}\ (t_4) & 5.5\mathrm{Cos}\ (t_4) \\ \mathrm{Sin}\ (t_4) & 0 & -\mathrm{Cos}\ (t_4) & 5.5\mathrm{Cos}\ (t_4) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$A_5 =$

$$\begin{bmatrix} \mathrm{Cos}\ (t_5) & -\mathrm{Sin}\ (t_5) & 0 & 0 \\ \mathrm{Sin}\ (t_5) & \mathrm{Cos}\ (t_5) & 0 & 0 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Transform Matrix is written as

T = A1*A2*A3*A4*A5,
We will get a 4*4 matrix where each element is equal to
r11=s1*s5 + c5*(c4*(c1*c2*c3 - c1*s2*s3) - s4*(c1*c2*s3 + c1*c3*s2))
r12=c5*s1 - s5*(c4*(c1*c2*c3 - c1*s2*s3) - s4*(c1*c2*s3 + c1*c3*s2))
r13=c4*(c1*c2*s3 + c1*c3*s2) + s4*(c1*c2*c3 - c1*s2*s3)
r14=d5*(c4*(c1*c2*s3 + c1*c3*s2) + s4*(c1*c2*c3 - c1*s2*s3)) + a2*c1*c2 + a4*c4*(c1*c2*c3 - c1*s2*s3) - a4*s4*(c1*c2*s3 + c1*c3*s2) - a3*c1*s2*s3 + a3*c1*c2*c3]
r21=c5*(c4*(c2*c3*s1 - s1*s2*s3) - s4*(c2*s1*s3 + c3*s1*s2)) - c1*s5
r22=c5*s1 - s5*(c4*(c1*c2*c3 - c1*s2*s3) - s4*(c1*c2*s3 + c1*c3*s2))
r23=c4*(c2*s1*s3 + c3*s1*s2) + s4*(c2*c3*s1 - s1*s2*s3)
r24=d5*(c4*(c2*s1*s3 + c3*s1*s2) + s4*(c2*c3*s1 - s1*s2*s3)) + a2*c2*s1 + a4*c4*(c2*c3*s1 - s1*s2*s3) - a4*s4*(c2*s1*s3 + c3*s1*s2) - a3*s1*s2*s3 + a3*c2*c3*s1]
r31=c5*(c4*(c2*s3 + c3*s2) + s4*(c2*c3 - s2*s3)) r32=-s5*(c4*(c2*s3 + c3*s2) + s4*(c2*c3 - s2*s3) r33=s4*(c2*s3 + c3*s2) - c4*(c2*c3 - s2*s3)
r34=d1 + a2*s2 - d5*(c4*(c2*c3 - s2*s3) - s4*(c2*s3 + c3*s2)) + a3*c2*s3 + a3*c3*s2 + a4*c4*(c2*s3 + c3*s2) + a4*s4*(c2*c3 - s2*s3)]

Part-C:

Forward Kinematics Validation

- Example-1

```
Command Window
>> p = ME598_GrpR8_FwdKin([80;55;-55;0;90])

p =

    5.078
   28.799
   11.243

>> |
```

- Example-2

```
Command Window

>> p = ME598_GrpR8_FwdKin([90;30;-30;-5;90])

p =

   1.9937e-15
        32.56
       14.739

>> 
```

- Example-3

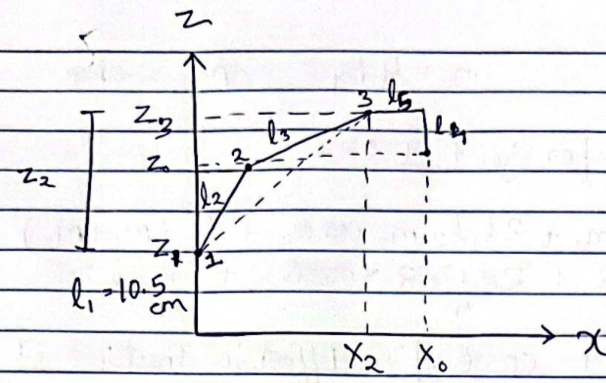
```
Command Window

>> p = ME598_GrpR8_FwdKin([-120;60;-90;25;90])

p =

      -12.738
      -22.062
       1.7973

>> 
```

- Example-3

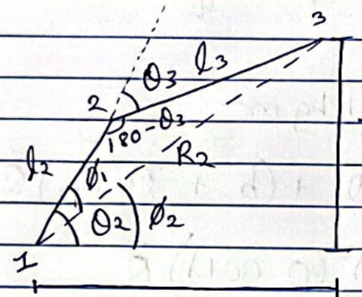
```
Command Window

>> p = ME598_GrpR8_FwdKin([40;15;-70;50;90])

p =

       20.749
       17.411
       2.9808

>> 
```

<u>Part-D:</u>

Inverse Kinematics



$l_2 = 13.5$ cm
$l_3 = 16$ cm
$l_5 = 5.5$ cm
$l_4 = 7$ cm



$$Z_2 = Z_0 + l_4 - l_1$$

$$R_2 = \sqrt{X_2^2 + Z_2^2}$$

$$X_2 = X_0 - l_5$$

Use cosine rule to find $(180 - Q_3)$ angle which in turn would give us Q_3

$$\cos(180 - Q_3) = \frac{l_2^2 + l_3^2 - R_2^2}{2 l_2 l_3} = -\cos(Q_3)$$

$$\therefore Q_3 = \cos\left(- \frac{(l_2^2 + l_3^2 - R_2^2)}{2 l_2 l_3} \right)$$

$$Q_2 = \phi_1 + \phi_2$$

$$\phi_2 = \tan^{-1}\left(\frac{Z_2}{X_2}\right)$$

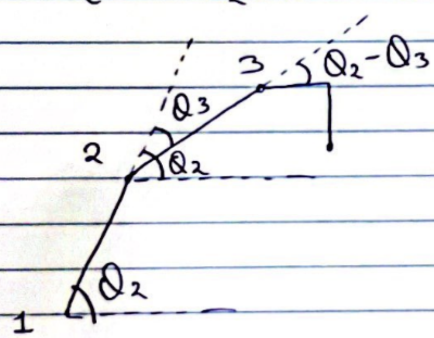
ϕ_1 is found using cosine rule

$$\cos(\phi_1) = \frac{l_2^2 + R_2^2 - l_3^2}{2 l_2 R_2}$$

$$\therefore \phi_1 = \cos^{-1}\left(\frac{l_2^2 + R_2^2 - l_3^2}{2 l_2 R_2}\right)$$

$$\theta_2 = \phi_1 + \phi_2$$

$$= \cos^{-1}\left(\frac{l_2^2 + R_2^2 - l_3^2}{2 l_2 R_2}\right) + \tan^{-1}\left(\frac{z_2}{x_2}\right)$$



$$\therefore \theta_4 = \theta_2 - \theta_3$$

And, θ_5 always remains zero.

<u>Part-E:</u>

Inverse Kinematics Validation

- **Example-1**

```
Command Window

>> q = ME598_GrpR8_InvKin([5.2;25;13])
Warning! End-effector2 beyond robot workspace!

q =

     []

fx q =
```

- **Example-2**

```
Command Window

>> q = ME598_GrpR8_InvKin([21.5;-7.83;11.3])

q =

      -20.011       85.604      -106.32       20.718            0

>>
>>
>>
```

- **Example-3**

```
Command Window

>>
>>
>> q = ME598_GrpR8_InvKin([21.23;17.81;-6.11])

q =

       39.994       26.266      -103.18       76.909            0

>>
```

<u>Part-F:</u>

Student Feedback

We can have a greater understanding of the material covered in the previous course thanks to this lab assignment, thus it is important that we go over everything we learned in order to complete the lab assignment correctly.

The major challenge we encounter is a lack of familiarity with the software. We have a lot of suggestions and can quickly put them on paper. Additionally, we are aware that programs like SolidWorks and Matlab might make it easier for us to finish the entire lab assignment. However, we are unable to quickly pick up on several of the software's features because we are unfamiliar with them. Because we can only write on paper, the report won't be strong enough. In addition, we don't have much experience with Matlab. Uncertainty over the syntax and features of the Matlab software is one of the challenges. To learn and use something requires some time. Thankfully, we managed to finish all of the lab homework.

In this lab assignment, we also overcame some of the difficulties in unified time communication. Because we have a team member who has not been able to come to the United States so far, we are not consistent when we have time. Our communication time and software need to be coordinated with each other, which gives us communication and cooperation have increased a certain degree of difficulty.

Appendix:

- %ME598_GrpR8 main

```
clear all
clc
syms t1 t2 t3 t4 t5 l1 l2 l3 l4 l5;
q = [t1;t2;t3;t4;t5];
c1 = cosd(t1);
c2 = cosd(t2);
c3 = cosd(t3);
c4 = cosd(t4);
c5 = cosd(t5);
s1 = sind(t1);
s2 = sind(t2);
s3 = sind(t3);
s4 = sind(t4);
s5 = sind(t5);
l1 = 10.5;
l2 = 13.5;
l3 = 16;
l4 = 5.5;
l5 = 7;
A1 = [c1 0 s1 0;s1 0 -c1 0;0 1 0 l1;0 0 0 1];
A2 = [c2 -s2 0 l2*c2;s2 c2 0 l2*c2;0 0 1 0;0 0 0 1];
A3 = [c3 -s3 0 l3*c3;s3 c3 0 l3*s3;0 0 1 0;0 0 0 1];
A4 = [c4 0 s4 l4*c4;s4 0 -c4 l4*s4;0 1 0 0;0 0 0 1];
A5 = [c5 -s5 0 0;s5 c5 0 0;0 0 1 l5;0 0 0 1];
T = A1*A2*A3*A4*A5;
function p = ME598_GrpR8_FwdKin(q);
end
function q = ME598_GrpR8_InvKin(p);
end
```

- function p = ME598_GrpR8_FwdKin(q)

```
%write the function for forward kinematics
l1 = 10.5;
l2 = 13.5;
l3 = 16;
l4 = 5.5;
l5 = 7;
t1 = q(1);
t2 = q(2);
t3 = q(3);
t4 = q(4);
t5 = q(5);
c1 = cosd(t1);
c2 = cosd(t2);
c3 = cosd(t3);
c4 = cosd(t4);
```

```
c5 = cosd(t5);
s1 = sind(t1);
s2 = sind(t2);
s3 = sind(t3);
s4 = sind(t4);
s5 = sind(t5);
if (t1<-130) && (t1>130)
   disp('Base Joint beyond robot workspace!');
   p = [];
   return;
end
if (t2<5) && (t2>95)
   disp('Shoulder Joint beyond workspace!');
   p =[];
   return;
end
if (t3>0) && (t3<-177)
   disp('Elbow Joint beyond workspace!');
   p = [];
   return;
end
if (t4<-5) && (t4>85)
   disp('Wrist pitch joint beyond worksapce!');
   p =[];
   return;
end
if (t5<-90) && (t5>90)
   disp('Wrist Twist angle beyond workspace!');
   p = [];
   return;
end
p1 = 7*cos((pi*t4)/180)*(cos((pi*t1)/180)*cos((pi*t2)/180)*sin((pi*t3)/180) +
cos((pi*t1)/180)*cos((pi*t3)/180)*sin((pi*t2)/180)) -
(11*cos((pi*t4)/180)*(cos((pi*t1)/180)*sin((pi*t2)/180)*sin((pi*t3)/180) -
cos((pi*t1)/180)*cos((pi*t2)/180)*cos((pi*t3)/180)))/2 -
(11*sin((pi*t4)/180)*(cos((pi*t1)/180)*cos((pi*t2)/180)*sin((pi*t3)/180) +
cos((pi*t1)/180)*cos((pi*t3)/180)*sin((pi*t2)/180)))/2 -
7*sin((pi*t4)/180)*(cos((pi*t1)/180)*sin((pi*t2)/180)*sin((pi*t3)/180) -
cos((pi*t1)/180)*cos((pi*t2)/180)*cos((pi*t3)/180)) +
(27*cos((pi*t1)/180)*cos((pi*t2)/180))/2 -
16*cos((pi*t1)/180)*sin((pi*t2)/180)*sin((pi*t3)/180) +
16*cos((pi*t1)/180)*cos((pi*t2)/180)*cos((pi*t3)/180);
p2 = 7*cos((pi*t4)/180)*(cos((pi*t2)/180)*sin((pi*t1)/180)*sin((pi*t3)/180) +
cos((pi*t3)/180)*sin((pi*t1)/180)*sin((pi*t2)/180)) +
(11*cos((pi*t4)/180)*(cos((pi*t2)/180)*cos((pi*t3)/180)*sin((pi*t1)/180) -
sin((pi*t1)/180)*sin((pi*t2)/180)*sin((pi*t3)/180)))/2 +
(27*cos((pi*t2)/180)*sin((pi*t1)/180))/2 -
(11*sin((pi*t4)/180)*(cos((pi*t2)/180)*sin((pi*t1)/180)*sin((pi*t3)/180) +
cos((pi*t3)/180)*sin((pi*t1)/180)*sin((pi*t2)/180)))/2 +
```

```
7*sin((pi*t4)/180)*(cos((pi*t2)/180)*cos((pi*t3)/180)*sin((pi*t1)/180) -
sin((pi*t1)/180)*sin((pi*t2)/180)*sin((pi*t3)/180)) +
16*cos((pi*t2)/180)*cos((pi*t3)/180)*sin((pi*t1)/180) -
16*sin((pi*t1)/180)*sin((pi*t2)/180)*sin((pi*t3)/180);
p3 = (27*cos((pi*t2)/180))/2 +
(11*cos((pi*t4)/180)*(cos((pi*t2)/180)*sin((pi*t3)/180) +
cos((pi*t3)/180)*sin((pi*t2)/180)))/2 -
7*cos((pi*t4)/180)*(cos((pi*t2)/180)*cos((pi*t3)/180) -
sin((pi*t2)/180)*sin((pi*t3)/180)) +
7*sin((pi*t4)/180)*(cos((pi*t2)/180)*sin((pi*t3)/180) +
cos((pi*t3)/180)*sin((pi*t2)/180)) +
(11*sin((pi*t4)/180)*(cos((pi*t2)/180)*cos((pi*t3)/180) -
sin((pi*t2)/180)*sin((pi*t3)/180)))/2 + 16*cos((pi*t2)/180)*sin((pi*t3)/180) +
16*cos((pi*t3)/180)*sin((pi*t2)/180) + 21/2;
p = [p1;p2;p3];

end
```

- `function q = ME598_GrpR8_InvKin(pos_mat)`

```
format short g
l1 = 10.5;
l2 = 13.5;
l3 = 16;
l4 = 7;
l5 = 5.5;
z_p = pos_mat(3) + l4 - l1;
x_p = pos_mat(1) - l5;
q(1) = atand(pos_mat(2)/pos_mat(1));
q(2) = atand( z_p / x_p ) + acosd( ((l2^2)+(z_p^2)+(x_p^2)-(l3^2)) /
(2*l2*sqrt(((z_p^2)+(x_p^2)))) );
q(3) = -acosd( -((l3^2)+(l2^2)-(z_p^2)-(x_p^2)) / (2*l3*l2) );
q(4) = - q(2) - q(3);
q(5) = 0;
if q(1) < -130 || q(1) > 130
   disp("Warning! End-effector1 beyond robot workspace!")
   q = []
   return
end
if q(2) < 5 || q(2) > 95
   disp("Warning! End-effector2 beyond robot workspace!")
   q = []
   return
end
if q(4) < -5 || q(4) > 85
   disp("Warning! End-effector3 beyond robot workspace!")
   q = []
   return
end
```