EENG 428 Introduction to Robotics Laboratory

Lab Session 3

Objective

In this Lab session, mathematical derivation examples for forward and inverse kinematic problems in simple coordinate systems are to be discussed with the assistance of the Matlab program to serve the purpose of this session.

Forward and Inverse Kinematics of Robots

Forward kinematic analysis is calculating the position and orientation of the end effector of a robot whose configuration is known. This means that all the link lengths and joint angles of the robot should be defined. On the other hand, the determination of the joint parameters that provide a desired position and orientation for the robot's end effector called *inverse kinematic analysis.*

In order to simplify the process, the position and orientation issues will be handled separately then we will combine the position and orientation analysis for a complete set of equations

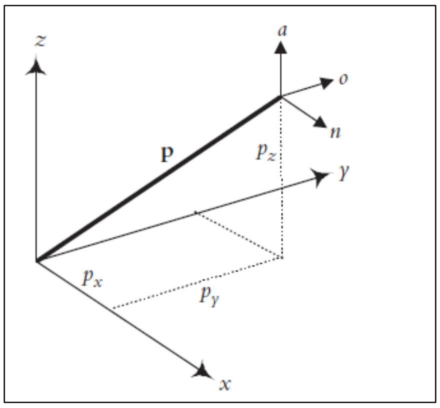
1- Forward and Inverse Kinematics: Position

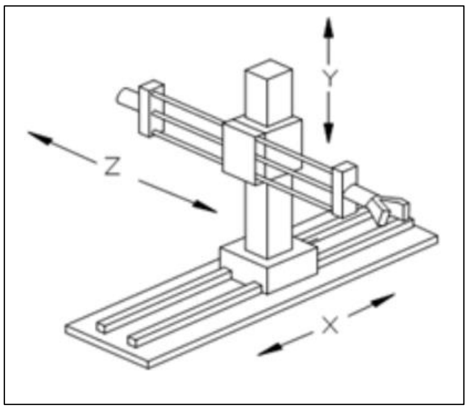
The position of the origin of a frame attached to a rigid body has three degrees of freedom, and therefore, the position can be completely defined by three pieces of information. As an example, we may position a point in space based on Cartesian coordinates, meaning there will be three linear movements relative to the x-, y-, and z-axes. Alternately, it may be accomplished through spherical coordinates, meaning there will be one linear motion and two rotary motions.

Prepared by: eyad.almasri@emu.edu.tr

1.1- Cartesian coordinates

A *Cartesian robot* is basically a Cartesian coordinate robot that is all actuators are linear placed in orthogonal manner and the positioning of the end effector is accomplished by moving the three linear joints along the three axes.





The position of the robot’s hand effector frame with respect to the base is given by

$$^{B}T_{H} = transl(p_x, p_y, p_z)$$

Both forward and inverse kinematics (for positions) are very simple to compute, the Matlab representation for forward and inverse kinematics for the Cartesian robot are as follows:

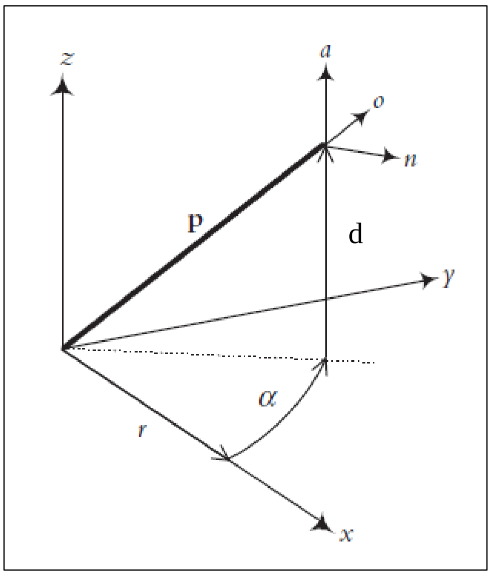
```
syms px py pz
px=1;
py=2;
pz=3;
T=transl(px,py,pz); % Forward kinematics
X=T(1:3,4); % Inverse kinematics
x=X(1)
Y=X(2)
z=X(3)
```

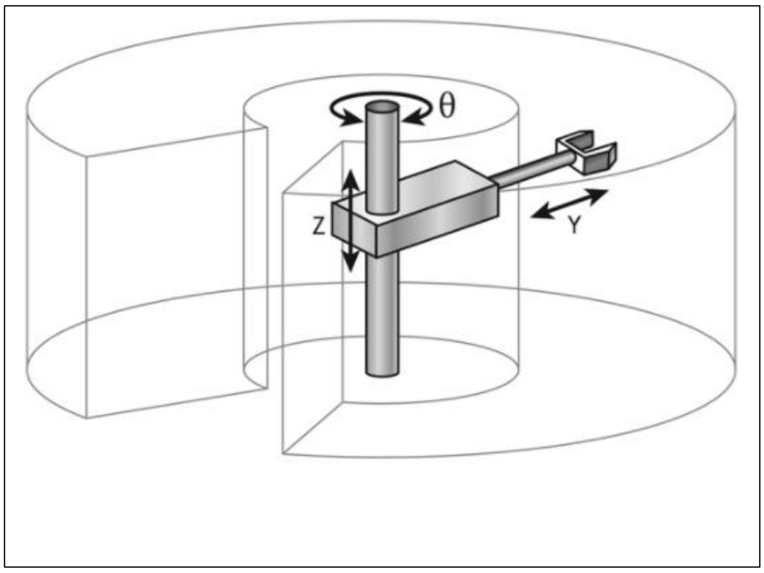
1.2- Cylindrical coordinates

A cylindrical coordinate system includes two linear translations and one rotation. The sequence is a translation of r along the x-axis, a rotation of a about the z-axis, and a translation of l along the z-axis. The position of the robot's hand effector frame with respect to the base is given by

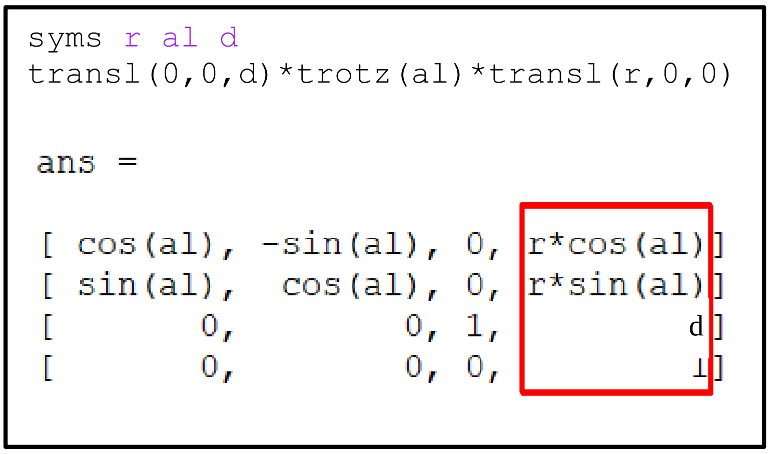
$$ {}^{B}T_{H} = transl\,(0,0,l\,)Rot\,(z\,,\alpha)transl\,(r\,,0,0) $$

Since all these transformations are all relative to the fixed frame, the total transformation is found by pre-multiplying by each matrix.





At this point, we are only interested in the position of the origin of the frame. You may restore the original orientation of the frame by rotating the n, o, a frame about the a-axis an angle of $-a$

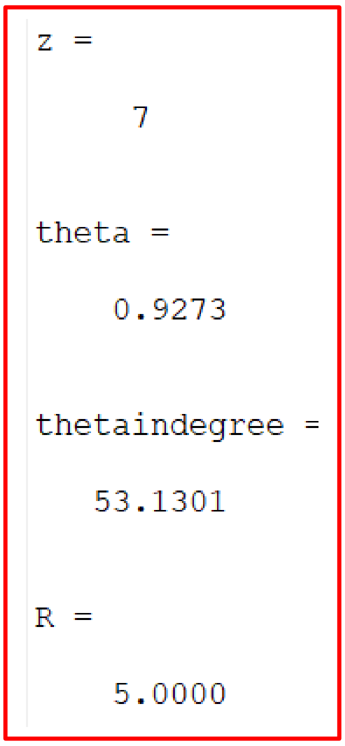


```
syms r al d
transl(0,0,d)*trotz(al)*transl(r,0,0)


 ans =

 [ cos(al), -sin(al), 0, r*cos(al)]
 [ sin(al),  cos(al), 0, r*sin(al)]
 [       0,        0, 1,         d]
 [       0,        0, 0,         1]
```

Example 1

Suppose we desire to place the origin of the hand frame of a cylindrical robot at $[3\ 4\ 7]^T$ Calculate joint variables of the robot

```
clc
clear all
 px=3;
 py=4;
 pz=7;

%Example1:
%inverse kinematics for
cylindrical coordinates
z= pz
theta=checkthequarter(py,px)
thetaindegree=57.2958*theta

if(px==0)
 R=py/sin(theta)
else
 R=px/cos(theta)
end
```

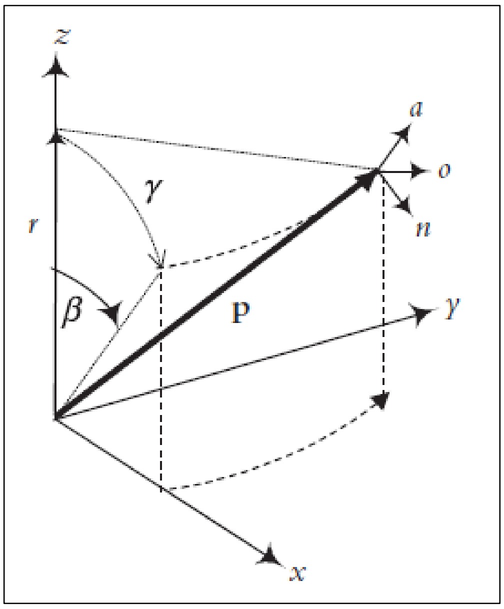


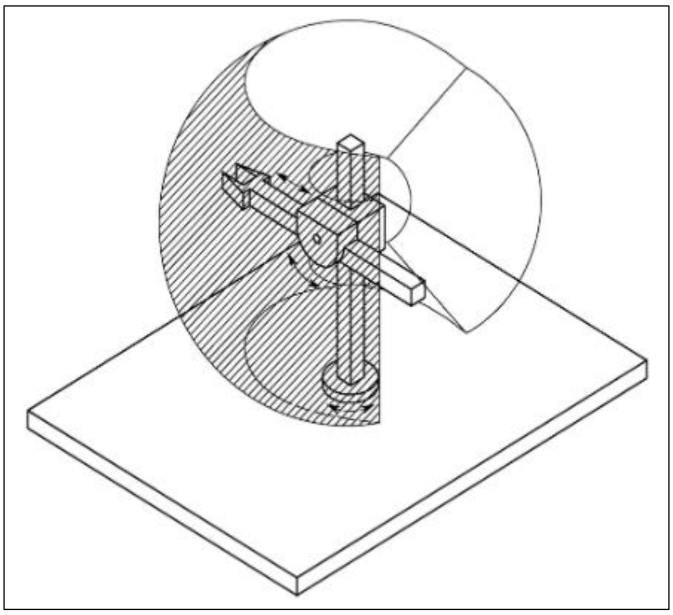
```
z =

     7


theta =

    0.9273


thetaindegree =

   53.1301


R =

    5.0000
```

```
function angle  = checkthequarter(x,y)
if(x>0&& y>0)
angle= atan(x/y);
thetaindegree=57.2958*angle;
end
if(x>0&& y<0)
angle= pi-atan(x/-y);
thetaindegree=57.2958*angle;
end
if(x<0&& y<0)
angle= pi+atan(x/y);
thetaindegree=57.2958*angle;
end
if(x<0&& y>0)
angle= -atan(-x/y);
thetaindegree=57.2958*angle;
end
end
```

1.3– Spherical coordinates

A spherical coordinate system consists of one linear motion and two rotations. The sequence is a translation of r along the z-axis, a rotation of β about the y-axis, and a rotation of γ about the z-axis





The total transformation caused by these three transformations can be found by pre-multiplying by each matrix since these transformations are all relative to the fixed frame

$$^{B}T_{H} = Rot(z,\gamma)Rot(y,\beta)transl(0,0,r)$$

```
syms gama beta r
trotz(gama)*troty(beta)*transl(0,0,r)
ans =

[ cos(beta)*cos(gama), -sin(gama), cos(gama)*sin(beta), r*cos(gama)*sin(beta)]
[ cos(beta)*sin(gama),  cos(gama), sin(beta)*sin(gama), r*sin(beta)*sin(gama)]
[          -sin(beta),          0,           cos(beta),           r*cos(beta)]
[                   0,          0,                   0,                     1]
```

At this point, we are only interested in the position of the origin of the frame.

Example 2

Suppose we desire to place the origin of the hand frame of a spherical robot at $[3\ 4\ 7]^T$ Calculate joint variables of the robot

```
%% Spherical coordinates
%% Inverse Kinematics
 clc
 clear all

 % example 2
 px=3;
 py=4;
 pz=7;

 %case 1 sin beta positive
 gama1= checkthequarter(py,px);
 gama1indegree=57.2958*gama1
 singama1=sin(gama1)
 cosgama1=cos(gama1)
 r_sin_beta1=px/cosgama1
 beta1= checkthequarter(r_sin_beta1,pz);
 beta1indegree=57.2958*beta1

 %case 2 sin beta negative
 gama2= checkthequarter(-py,-px);
 gama2indegree=57.2958*gama2
 singama2=sin(gama2)
 cosgama2=cos(gama2)
 r_sin_beta2=px/cosgama2
 beta2= checkthequarter(r_sin_beta2,pz);
 beta2indegree=57.2958*beta2

 if(pz~=0)
 R=pz/cos(beta1)
 else if(py~=0)
 R=py/(sin(beta1)*sin(gama1))
 else if(px~=0)
 R=px/(sin(beta1)*cos(gama1))
 end
 end
 end
```

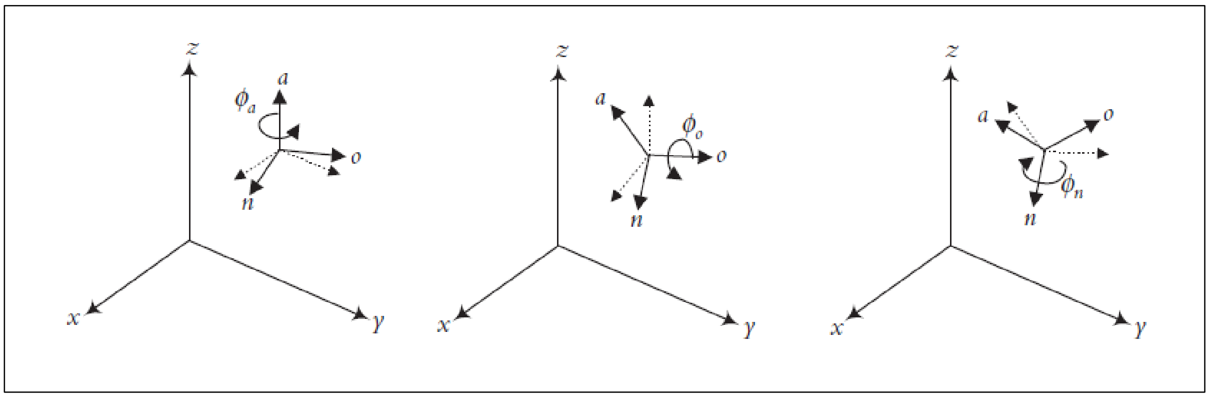
```
gama1indegree =53.1301
singama1 =0.8000
cosgama1 = 0.6000
r_sin_beta1 =  5.0000
beta1indegree =  35.5377
gama2indegree =233.1302
singama2 = -0.8000
cosgama2 = -0.6000
r_sin_beta2 =  -5.0000
beta2indegree =-35.5377
R = 8.6023
```

2-**Forward and Inverse Kinematics: Orientation**

Suppose the moving frame attached to the hand of the robot has already moved to a desired position in Cartesian, cylindrical or spherical coordinates. The next step will be to rotate the frame appropriately in order to achieve a desired orientation without changing its position. This can only be accomplished by rotating about the current frame axes rotations about the reference frame axes will change the position.

2.1- Roll, Pitch and Yaw Angles

This is a sequence of three rotations about current a-, o-, and n-axes respectively. Therefore, all matrices related to the orientation change due to RPY will be post-multiplied. Any orientation in 3D space can be achieved by these three sequential rotations



$$^{B}R_{H} = Rot(z, a1)Rot(y, a2)Rot(x, a3)$$

$$^{B}R_{H} = \begin{bmatrix} Ca1\,Ca2 & Ca1\,Sa2\,Sa3 - Sa1\,Ca3 & Ca1\,Sa2\,Ca3 + Sa1\,Sa3 \\ Sa1\,Ca2 & Sa1\,Sa2\,Sa3 + Ca1\,Ca3 & Sa1\,Sa2\,Ca3 - Ca1\,Sa3 \\ -Sa2 & Ca2\,Sa3 & Ca2\,Ca3 \end{bmatrix}$$

Forward kinematics: if RPY angles are given, the orientation of the last frame can be directly computed, and it is unique

```
syms a1 a2 a3
Trpy=trotz(a1)*troty(a2)*trotx(a3)
```

Inverse kinematics: if the desired orientation is given, the values of the RPY angles can be calculated to achieve the same orientation. Usually, there are two set of solutions that give the same orientation. You can check the derivation of the following inverse kinematics equations in [1].

$$\begin{cases} a1 = ATAN2(ny, nx) \quad and \quad ATAN2(-ny, -nx) \\ a2 = ATAN2(-nz, (nxCa1 + nySa1)) \\ a3 = ATAN2((-ayCa1 + axSa1), (oyCa1 - oxSa1)) \end{cases}$$

Example 3:

The desired final Orientation of the hand of a RPY robot is given below. Find the necessary RPY angles

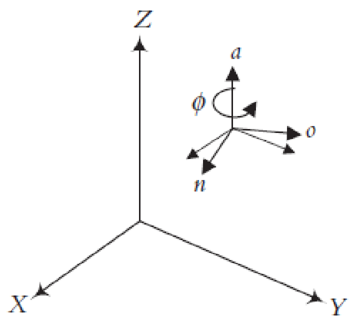
$$\begin{bmatrix} 0.354 & -0.674 & 0.649 \\ 0.505 & 0.722 & 0.475 \\ 0.788 & 0.160 & 0.595 \end{bmatrix}$$

```
clear all
close all
syms a3 a2 a1
T=[0.354 -0.674 0.649;0.505 0.722 0.475;-0.788 0.160 0.595 ]
Trpy=trotz(a1)*troty(a2)*trotx(a3)

S1_of_a1=checkthequarter(T(2,1),T(1,1))
S1_of_a1_in_degree=57.2958*S1_of_a1
S2_of_a1=checkthequarter(-T(2,1),-T(1,1))
S2_of_a1_in_degree=57.2958*S2_of_a1

S1_of_a2=checkthequarter(-T(3,1),(
T(1,1)*cos(S1_of_a1)+T(2,1)*sin(S1_of_a1)))
S1_of_a2_in_degree=57.2958*S1_of_a2
S2_of_a2=checkthequarter(-T(3,1),(
T(1,1)*cos(S2_of_a1)+T(2,1)*sin(S2_of_a1)))
S2_of_a2_in_degree=57.2958*S2_of_a2

S1_of_a3=checkthequarter(-
T(2,3)*cos(S1_of_a1)+T(1,3)*sin(S1_of_a1),T(2,2)*cos(S1_of_a1)-
T(1,2)*sin(S1_of_a1))
S1_of_a3_in_degree=57.2958*S1_of_a3
S2_of_a3=checkthequarter(-
T(2,3)*cos(S2_of_a1)+T(1,3)*sin(S2_of_a1),T(2,2)*cos(S2_of_a1)-
T(1,2)*sin(S2_of_a1))
S2_of_a3_in_degree=57.2958*S2_of_a3
```

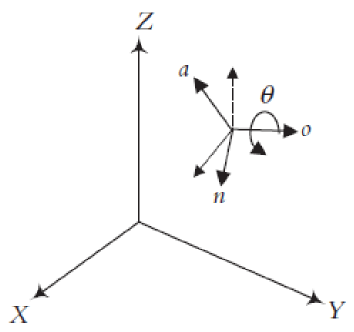
```
S1_of_a1_in_degree = 54.9699
S2_of_a1_in_degree =234.9700
S1_of_a2_in_degree = 51.9520
S2_of_a2_in_degree = 128.0481
S1_of_a3_in_degree = 14.9918
S2_of_a3_in_degree = 194.9919
```

2.2- Euler Angles

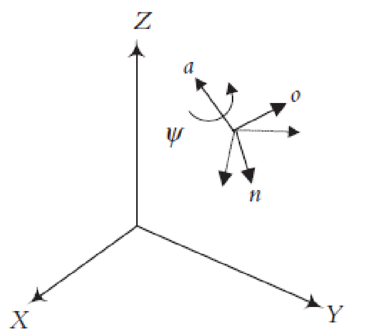
Euler angles are very similar to RPY, except that the last rotation is also about the current a axis



Rotation of ϕ about the a-axis



Rotation of θ about the o-axis



Rotation of ψ about the a-axis

Homework:

Given the following transformation matrix, the orientation part written based on Euler Angles.

$$^{R}T_{P} = \begin{bmatrix} 0.354 & -0.674 & 0.649 & 0 \\ 0.505 & 0.722 & 0.475 & 0 \\ -0.788 & 0.160 & 0.595 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Find a way to give all the sets of the Euler angles that achieve the same orientation.

(You can use the Matlab or you can solve it by hand. In both situations, you have to clear all the details that lead you to your answer)

<u>Direct answers are not accepted</u>

<u>Ready Function is not accepted solution</u>

3- **Forward and Inverse Kinematics: Position and Orientation**

The matrix representing the final location and orientation of the robot is a combination of the above, depending on which coordinates are used. If a robot is made of a Cartesian and an RPY set of joints, then the location and the final orientation of the frame relative to the reference frame will be the product of the two matrices representing the Cartesian position change and the RPY. The robot may be represented by:

$$^{R}T_{H} = T_{cart}(p_x, p_y, p_z) \times \mathrm{RPY}(\phi_a, \phi_o, \phi_n)$$

Homework:

Suppose that a robot is made of a Cartesian and RPY combination of joints. Find the necessary RPY angles and the displacement of the Cartesian coordinates to achieve the following

$$T = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 4 \\ 0.369 & 0.819 & 0.439 & 6 \\ -0.766 & 0 & 0.643 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

References:

1- Niku, S. (2010). *Introduction to robotics*. John Wiley & Sons.