Name: Girish S **Roll No: AM.EN.U4AIE22044**

INTRODUCTION TO AI ROBOTICS

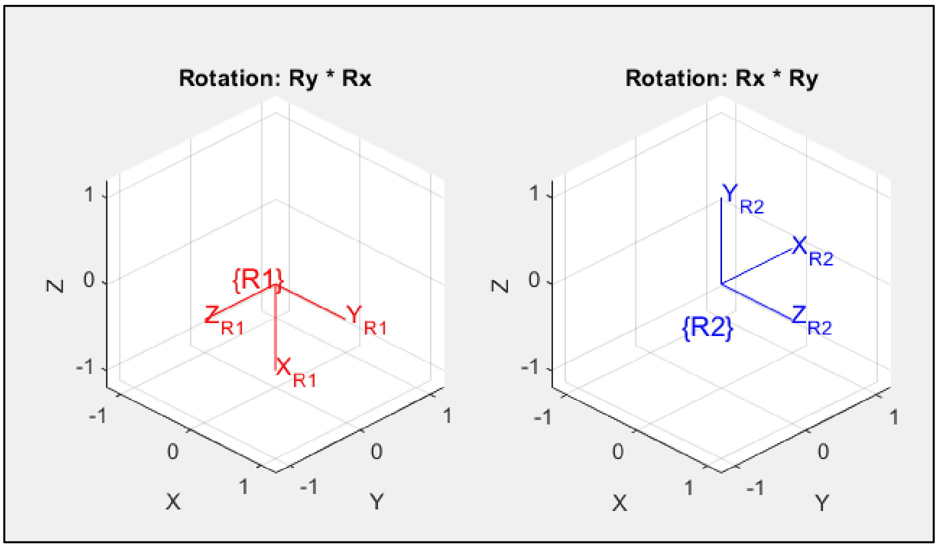
Labsheet – 4

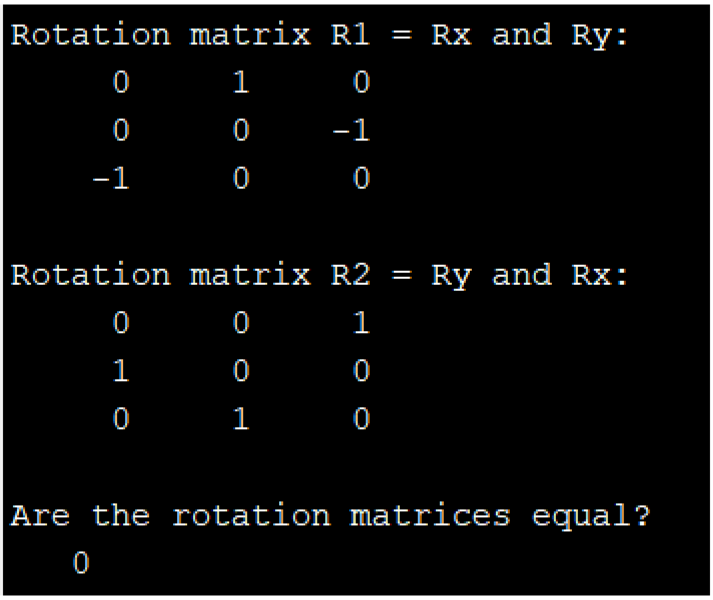
Q1) Justify the statement that "*Rotations are non-commutative in 3D*".

Instruction: Show the Matlab operations with example. You can put snapshots in your answer sheet and give relevant explanation.

Refer Text book Robotics Vison and Control by Peter Corke page 29-35

```
Rx = rotx(90, 'deg');
Ry = roty(90, 'deg');

R1 = Ry * Rx;
R2 = Rx * Ry;

disp('Rotation matrix R1 = Rx and Ry:');
disp(R1);

disp('Rotation matrix R2 = Ry and Rx:');
disp(R2);

isEqual = isequal(round(R1, 10), round(R2, 10));
disp('Are the rotation matrices equal?');
disp(isEqual);

figure;
subplot(1,2,1);
trplot(R1, 'frame', 'R1', 'color', 'r');
title('Rotation: Ry * Rx');
view(45, 30);

subplot(1,2,2);
trplot(R2, 'frame', 'R2', 'color', 'b');
title('Rotation: Rx * Ry');
view(45, 30);
```





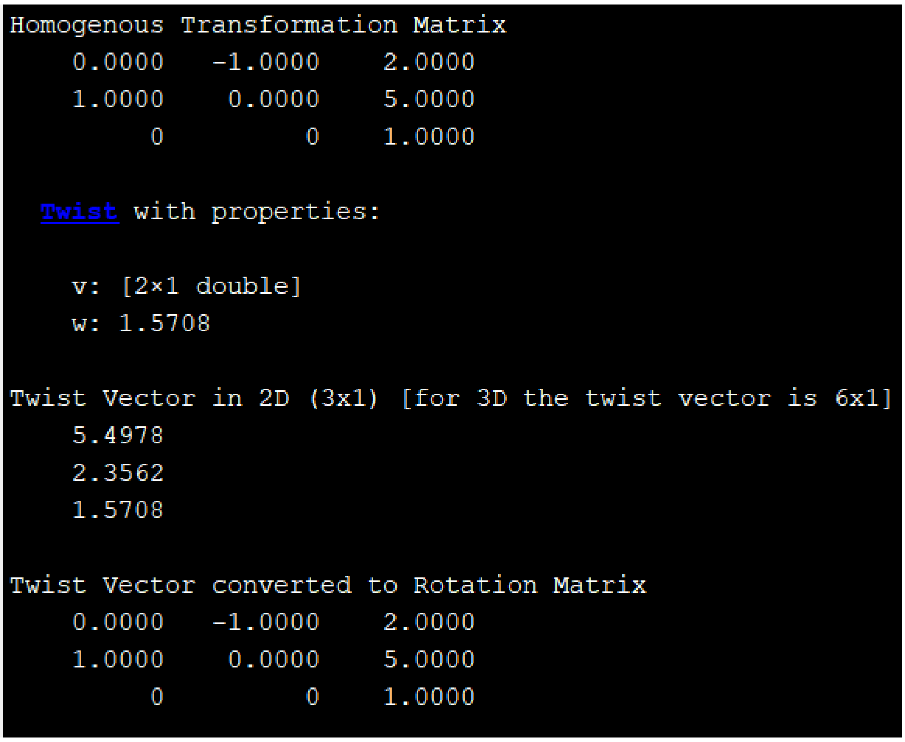
```
Rotation matrix R1 = Rx and Ry:
     0     1     0
     0     0    -1
    -1     0     0

Rotation matrix R2 = Ry and Rx:
     0     0     1
     1     0     0
     0     1     0

Are the rotation matrices equal?
   0
```

Q2) Describe twist in 2D.

Use matlab functions to show how to regenerate homogeneous transformations

A Twist is a 6-vector consisting of a 3-vector expressing the angular velocity and a 3-vector expressing the linear velocity. Both of these are written in coordinates of the same frame, and the linear velocity refers to the linear velocity of a point at the origin of that frame. Both the body twist and the spatial twist represent the same motion, just in different coordinate frames. The body twist is not affected by the choice of the space frame, and the spatial twist is not affected by the choice of the body frame.

```
R = transl2(2,5) * trot2(90,"deg");
disp('Homogenous Transformation Matrix');
disp(R);

t=Twist(R);
disp(t);
disp('Twist Vector in 2D (3x1) [for 3D the twist vector is 6x1]');
disp(t.S);
disp('Twist Vector converted to Rotation Matrix');
disp(t.T);
```

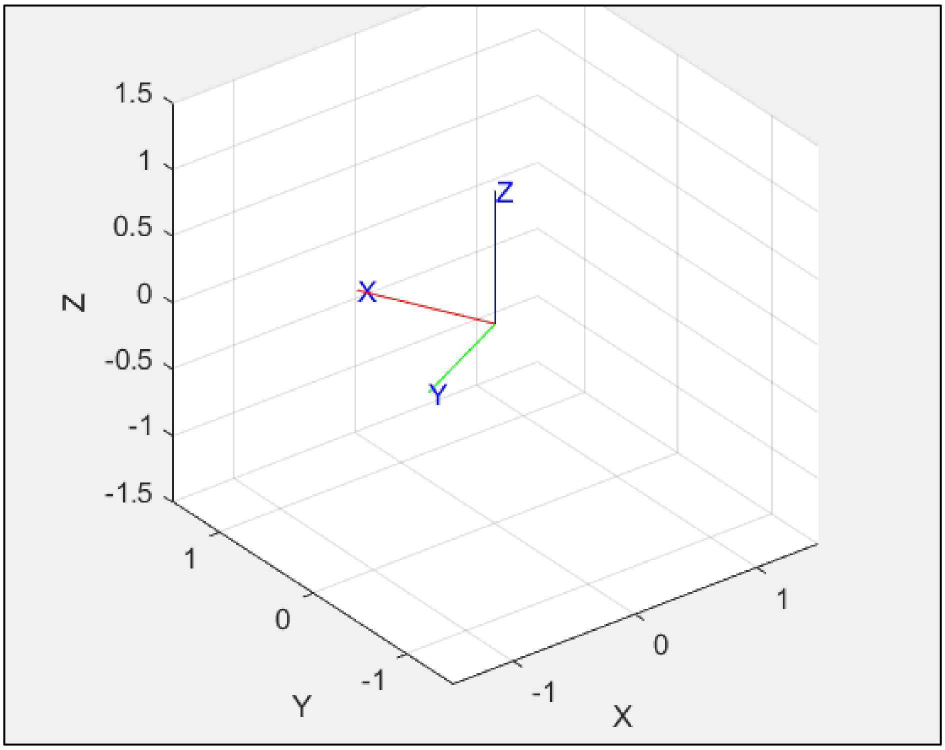


```
Homogenous Transformation Matrix
    0.0000   -1.0000    2.0000
    1.0000    0.0000    5.0000
         0         0    1.0000

  Twist with properties:

    v: [2×1 double]
    w: 1.5708

Twist Vector in 2D (3x1) [for 3D the twist vector is 6x1]
    5.4978
    2.3562
    1.5708

Twist Vector converted to Rotation Matrix
    0.0000   -1.0000    2.0000
    1.0000    0.0000    5.0000
         0         0    1.0000
```

Q3) Demonstrate the Toolbox function tranimate which animates a rotation using Matlab.

Refer Text book Robotics Vison and Control by Peter Corke page 29-35

```
T0 = eye(4);
T1 = trotz(90);
figure;
axis([-1 1 -1 1 -1 1]);
view(3);
tranimate(T0, T1, 'frames', 50, 'rgb');
```


1.5
1
0.5
0
-0.5
-1
-1.5
Z
Z
X
Y
1
0
-1
Y
-1
0
1
X