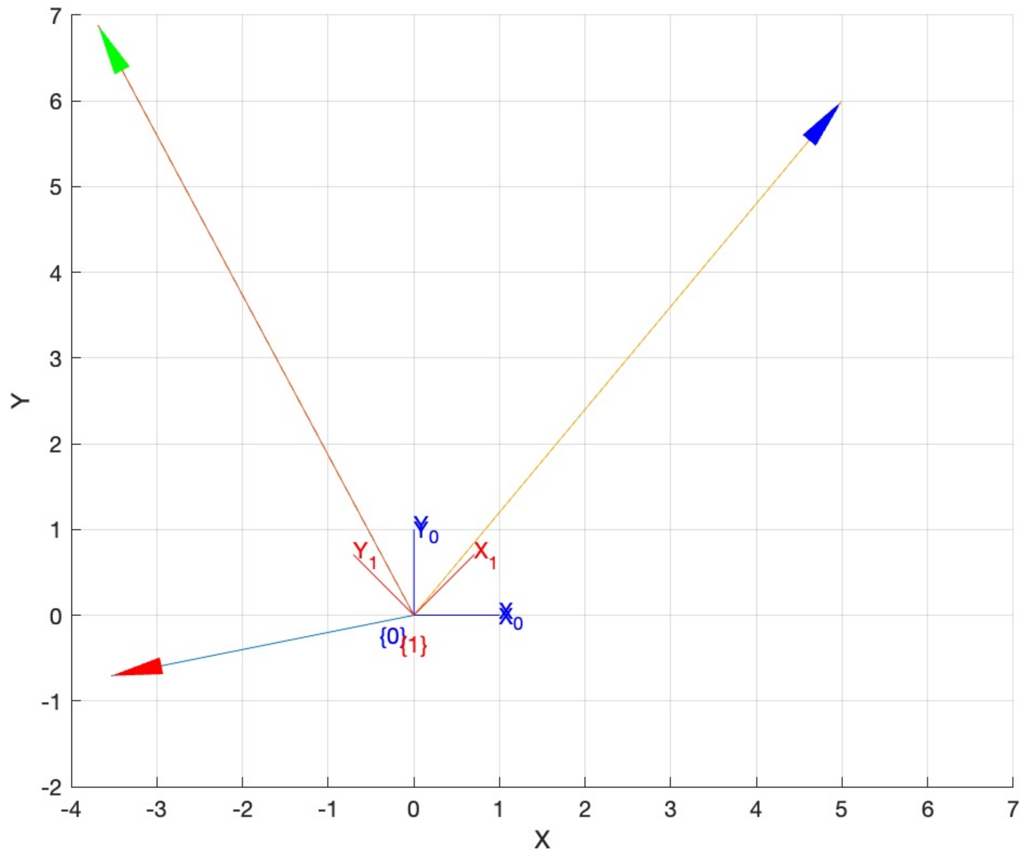
Answer Sheet

Index No: 210204R

1. MATLAB code for 3.1 ~ 3.5.

```
%3.1
trplot2;
axis([-4, 7, -2, 7]);
grid;

%3.2
hold on;
plot_arrow([0 0]', [5 6]', 'b');

%3.3
theta = 45 * pi / 180;
R1_0 = rot2(theta);    R0_1 = R1_0';

trplot2(eye(2), 'frame', '0');
axis([-4 7 -2 7]);
grid on;

tranimate2(R1_0, 'color', 'r', 'frame', '1', 'fps', 20, 'nsteps', 100, 'axis', [-4 7 -2 7]);

p = [5; 6];
p1 = R0_1 * p;

disp('Coordinates of point p in frame {1}:'); disp(p1);

%3.4
q_in_1 = [-3; 2];
q_in_0 = R1_0 * q_in_1;

disp('Coordinates of point q in frame {0}:'); disp(q_in_0);
plot_arrow([0, 0], q_in_0, 'r');


%3.5
theta_68 = 68 * pi / 180;
R_68_0 = rot2(theta_68);
r = R_68_0 * p;

disp('Coordinates of point r:');
disp(r)

plot_arrow([0, 0], r, 'g');
hold on;
axis([-4, 7, -2, 7]);
grid on;
```

2. Final output MATLAB figure for the operations in 3.1 ~ 3.5.



3. p^1 for 3.3:

(7.7782, 0.7071)

```
Coordinates of point p in frame {1}:
    7.7782
    0.7071
```

4. R_1^0 for 3.7.

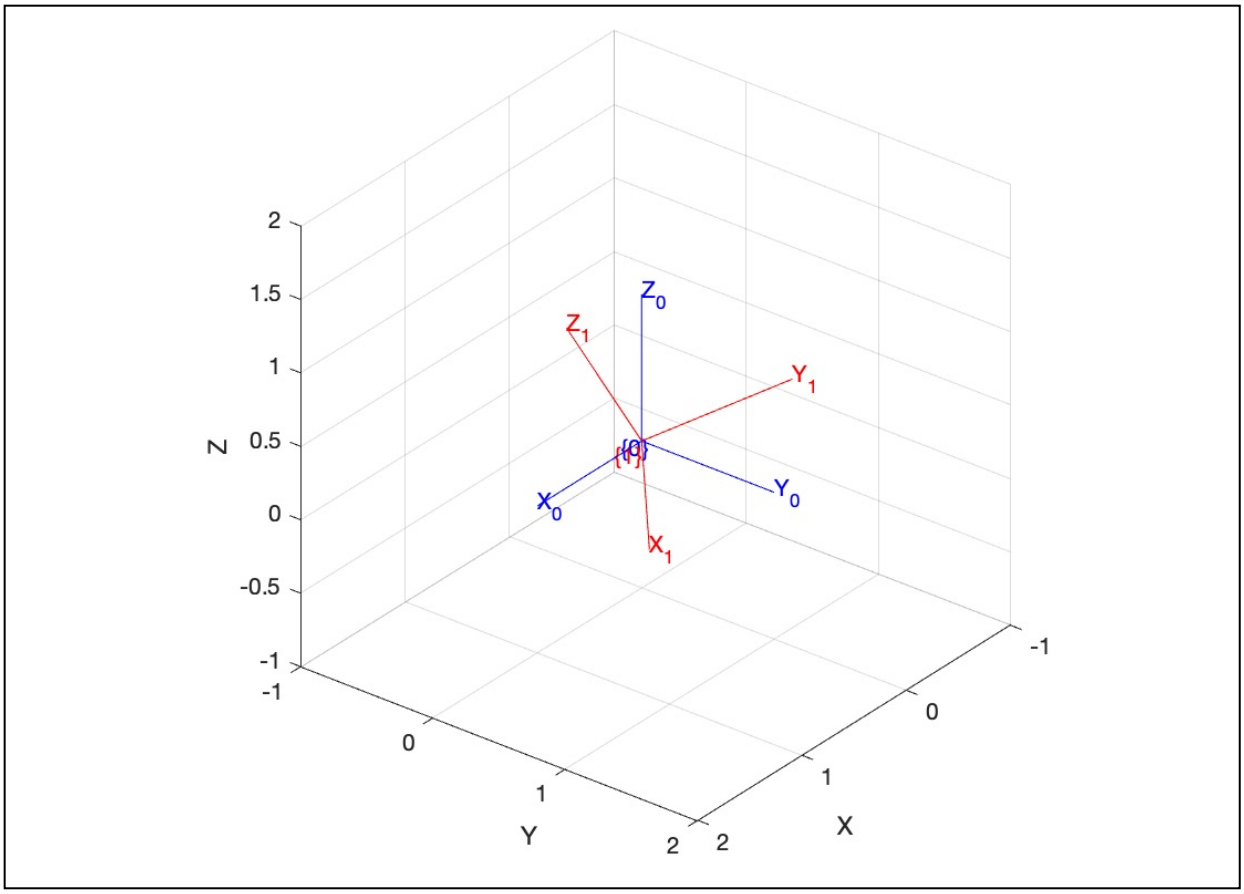
```
Final Rotation Matrix R1_0:
    0.7424   -0.5198    0.4226
    0.6436    0.7285   -0.2346
   -0.1859    0.4462    0.8754
```

5. MATLAB code for 3.6 ~ 3.9.

```
%3.6
figure;
trplot(eye(3), 'frame', '0');
axis([-1 2 -1 2 -1 2]);
grid on;
view(3)

%3.7
theta_x = 15 * pi / 180;
theta_y = 25 * pi / 180;
theta_z = 35 * pi / 180;

R_x = rotx(theta_x);
R_y = roty(theta_y);
R_z = rotz(theta_z);

disp('Rotation about X-axis 15 deg:');
disp(R_x);
disp('Rotation about Y-axis 25 deg:');
disp(R_y);
disp('Rotation about Z-axis 35 deg:');
disp(R_z);

% Compute the final rotation matrix R1_03D
R1_03D = R_x * R_y * R_z;
disp('Final Rotation Matrix R1_0:');
disp(R1_03D);

tranimate(eye(3), R_x, 'fps', 20, 'nsteps', 50, 'rgb', 'axis', [-1 1 -1 1 -1 1],'cleanup');
tranimate(R_x, R_x*R_y, 'fps', 20, 'nsteps', 50, 'rgb', 'axis', [-1 1 -1 1 -1 1],'cleanup');
tranimate(R_x*R_y, R_x*R_y*R_z, 'fps', 20, 'nsteps', 50, 'rgb', 'axis', [-1 1 -1 1 -1 1]);

figure;
trplot(eye(3), 'frame', '0', 'color', 'b');  % Blue frame for {0}
hold on;

trplot(R1_03D, 'frame', '1', 'color', 'r');  % Red frame for {1}

axis([-1 2 -1 2 -1 2]);
grid on;
view(3)
```

```
%3.9
R = [0.8138 0.0400 0.5798;
     0.2962 0.8298 -0.4730;
    -0.5000 0.5567 0.6634];

rpy_angles = tr2rpy(R, 'zyx');

roll = rad2deg(rpy_angles(1));  pitch = rad2deg(rpy_angles(2)); yaw = rad2deg(rpy_angles(3));

% Display the results
disp('Roll (ψ) in degrees:');
disp(roll);
disp('Pitch (θ) in degrees:');
disp(pitch);
disp('Yaw (φ) in degrees:');
disp(yaw);

% Confirm by doing the reverse operation: Calculate rotation matrix from RPY angles
R_confirm = rpy2r(deg2rad(roll), deg2rad(pitch), deg2rad(yaw), 'zyx');
disp('Confirmed rotation matrix from RPY angles:');
disp(R_confirm);

% Display the confirmed rotation matrix from the basic matrices
disp('Confirmed rotation matrix from basic matrices');
% Create the 3x3 rotation matrices
R_x = rotx(rpy_angles(1));  % Rotation about X-axis
R_y = roty(rpy_angles(2));  % Rotation about new Y-axis
R_z = rotz(rpy_angles(3));  % Rotation about new Z-axis
R_matrices =R_z*R_y*R_x;
disp(R_matrices);
```

6. Final output MATLAB figure for the operations in 3.6 ~ 3.9.



7. Default roll-pitch-yaw angle definition for the toolbox.

8. For 3.9,
ψ: 40.0021 θ: 29.9999 φ: 20.0001

```
Roll (ψ) in degrees:
   40.0021

Pitch (θ) in degrees:
   29.9999

Yaw (φ) in degrees:
   20.0001

Confirmed rotation matrix from RPY angles:
    0.8138    0.0400    0.5798
    0.2962    0.8298   -0.4731
   -0.5000    0.5567    0.6634

Confirmed rotation matrix from basic matrices
    0.8138    0.0400    0.5798
    0.2962    0.8298   -0.4731
   -0.5000    0.5567    0.6634
```