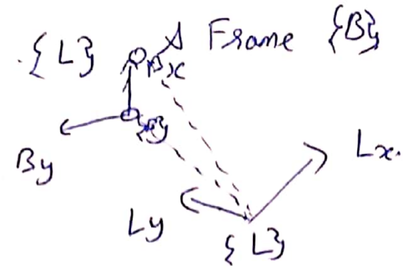
Assignment - 2

1.) Given,

Frame $\{L\}$ Frame $\{B\}$



$$ {}^{L}P_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}, \quad {}^{L}P_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} $$

1.a

$$ {}^{L}T_B = \begin{bmatrix} {}^{L}X_B & {}^{L}Y_B & {}^{L}Z_B & {}^{L}P_{1B} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

x-axis of $\{B\}$ $\{{}^{L}X_B\}$ $= \overrightarrow{P_1 P_2} = \dfrac{P_2 - P_1}{\text{norm}(\overrightarrow{P_1 P_2})}$

$$ = \begin{bmatrix} \dfrac{x_2 - x_1}{\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}} & \dfrac{(y_2 - y_1)}{\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}} & 0 \end{bmatrix}^T $$

$$ = \begin{bmatrix} \dfrac{5}{\sqrt{5^2+5^2}} & \dfrac{5}{\sqrt{5^2+5^2}} & 0 \end{bmatrix}^T $$

$$ = \begin{bmatrix} \dfrac{5}{5\sqrt{2}} & \dfrac{5}{5\sqrt{2}} & 0 \end{bmatrix}^T = \begin{bmatrix} \dfrac{1}{\sqrt{2}} & \dfrac{1}{\sqrt{2}} & 0 \end{bmatrix}^T $$

z-axis of B $\{{}^{L}Z_B\}$ $= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$

y-axis of B $= {}^{L}Z_B \times {}^{L}X_B$

$$ = \begin{bmatrix} i & j & k \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} = i\left(-\frac{1}{\sqrt{2}}\right) - j\left(-\frac{1}{\sqrt{2}}\right) $$

$$ = \begin{bmatrix} -\dfrac{1}{\sqrt{2}} & \dfrac{1}{\sqrt{2}} & 0 \end{bmatrix}^T $$

Now to get L_{TB}

$$L_{T_B} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 5 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 15 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.71 & -0.71 & 0 & 5 \\ 0.71 & 0.71 & 0 & 15 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1. \end{bmatrix} \quad -①$$

Equation ① represents the solution for.

1.a

1.b Given, $x_1 = x_2$. $y_1 > y_2$.

$$L_{x_B} = \left[\frac{0}{\sqrt{0 + \sqrt{(y_2 - y_1)^2}}} \quad \frac{y_2 - y_1}{\sqrt{0 + (y_2 - y_1)^2}} \quad 0 \right]^T$$

$$= \left[0 \quad \frac{y_2 - y_1}{\sqrt{(y_2 - y_1)^2}} \quad 0 \right]^T$$

Since $y_1 > y_2$.

$$L_{xB} = \begin{bmatrix} 0 & -1. & 0 \end{bmatrix}^T$$

$$L_{zB} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$L_{y_B} = \begin{bmatrix} i & j & k. \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} = i(0+1) - j(0-0) + k[0-0]$$

$$= i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T.$$

$$L_{T_B} = \begin{bmatrix} 0 & 1 & 0 & 5 \\ -1 & 0 & 0 & 15 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad - (2)$$

Equation (2) represents the solution for 1.b.

2.)

Given To compute the rotational matrix of IMU.

2.a

$$a_{IMU} = \begin{bmatrix} 0 \\ -3.53 \\ 9.15 \end{bmatrix}, \quad g = 9.81\ m/s^2.$$

$$\theta_y = 0^\circ$$

Now

$$\sin\theta_p = -\frac{a_x}{g} = 0.$$

$$\sin\theta_p = 0$$

$$\theta_p = \sin^{-1}(0)$$

$$\boxed{\theta_p = 0^\circ} \quad -(1)$$

$$\tan\theta_r = \frac{a_y}{a_z} = -\frac{3.53}{9.15}$$

$$\theta_r = \tan^{-1}\left(\frac{-3.53}{9.15}\right)$$

$$\boxed{\theta_r = -21.1^\circ}$$

Orthonormal rotation matrix R

$$R = \begin{bmatrix} \cos\theta_y & -\sin\theta_y & 0 \\ \sin\theta_y & \cos\theta_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_p & 0 & \sin\theta_p \\ 0 & 1 & 0 \\ -\sin\theta_p & 0 & \cos\theta_p \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_r & -\sin\theta_r \\ 0 & \sin\theta_r & \cos\theta_r \end{bmatrix}$$

Now $\theta_y = 0^\circ$, $\theta_p = 0^\circ$, $\theta_r = -21.1^\circ$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.93 & 0.36 \\ 0 & -0.36 & 0.93 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.93 & 0.36 \\ 0 & -0.36 & 0.93 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.93 & 0.36 \\ 0 & -0.36 & 0.93 \end{bmatrix}.$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.933 & 0.36 \\ 0 & -0.36 & 0.933 \end{bmatrix}. \quad — (1)$$

(1) is solution for $\underline{\underline{z}}$.a

2.b :

Please find the Matlab code below and the results followed by the code.
The R Matrix in the following code is considered from the previous question

CODE WITHOUT NORMALIZATION:

```
% R- Value taken from answer of 2.a
R = [1 0 0;0 0.933 0.36;0 -0.36 0.933];

% Given omega_IMU in 2.b
omega_IMU = [0.7 0.8 0];

%loop five times to get the updated R
for m= 1:5
        R = R + R*0.06*skew(omega_IMU);
end

% Print the R matrix
R

% To get the Determinant of R
det(R)
```

RESULT FOR THE ABOVE CODE:

```
R =

   0.9770    0.0201    0.2380
  -0.0669    0.9916    0.1510
  -0.2293   -0.1593    0.9701


ans =

        1.0206
```

CODE WITH NORMALIZATION:

```
% R-Matrix from 2.a
R = [1 0 0;0 0.933 0.36;0 -0.36 0.933];

%loop over 5 times and normalize the matrices
for m=1:5
        % c3' = c3
        %c1' =  c2 * c3'
        R(1:3,1) = cross(R(1:3,2),R(1:3,3));

        %c2' = c3' * c1'
        R(1:3,2) = cross(R(1:3,3),R(1:3,1));

        %c1'' = c1'/norm(c1')
        R(1:3,1) = R(1:3,1)/norm(R(1:3,1));

        %c2'' = c2'/norm(c2')
        R(1:3,2) = R(1:3,2)/norm(R(1:3,2));

        %c3'' = c3/norm(c3')
        R(1:3,3) = R(1:3,3)/norm(R(1:3,3));

end

%display the R matrix
R

% display the determinant of R
det(R)
```

RESULT FOR ABOVE CODE:

```
R =
   1.0000         0          0
        0    0.9330     0.3600
        0   -0.3600     0.9330

ans =
        1
```

Summary : There are some small changes in rotation matrix after normalization. But the major update after normalization is that the magnitude of Rotation matrix will be exactly 1 after normalization.

3.

Code for LSPB curve and Results:

```
clear all;
tf = 4; % Given time limit
theta_f = 65; % final position(in degrees)
theta_o = -10; % initial position(in degrees)
a = 30; % Acceleration of theta (ind deg/sec^2)

tb = (tf/2) - ((sqrt((a^2*tf^2)-(4*a*(theta_f-theta_o)))/(2*a))); % (The initial blend time)

tbo = tf - tb; % final time minus the beldn time, time where the blend region ends

theta_b =  theta_o+ 0.5*a* tb^2; % calculation of theta in blend time tb
theta_b_vel = a*tb; % vleocity at the blend time tb

t = 0:0.01:tf;

r1 = (t < tb);  % region 1 with increasing velocity

r2 = (t>=tb) & (t<tbo); % region 2 with constant velocity

r3 = (t>=tbo); % region 3 with decreasing velocity

pos= zeros(size(t));
vel= zeros(size(t));
acc= zeros(size(t));

% Calculation of position in the three regions
pos(r1) = theta_o + (0.5*a*t(r1).^2 );
pos(r2) = theta_o - (0.5*a*tb.^2) + (theta_b_vel*t(r2));
pos(r3) = theta_f - (0.5*a*(tf-t(r3)).^2);
%plot(pos)

%Calculations of Velocity in the three regions
vel(r1) = a*t(r1);
vel(r2) = theta_b_vel;
vel(r3) = a*(tf-t(r3));
%plot(vel)

%Calculations of acceleration in the three regions
acc(r1) = a;
acc(r2) = 0.0;
```

```
acc(r3) = -a;

%Plot the figure
figure
subplot(3,1,1)
plot(pos)
title('Position')
subplot(3,1,2)
plot(vel)
title('Velocity')
subplot(3,1,3)
plot(acc)
title('Acceleration')
```