**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')
```