ME131 Lab7 Deliverables

Task 1:

1. Rotation matrices in `me131_lab7_script.m`

```
%% ****** Deliverable 1*****
%% Define Rotation matrices
Ryaw = @(yaw) [cos(yaw), -sin(yaw), 0;
               sin(yaw), cos(yaw),  0;
               0,        0,         1];

Rpitch = @(pitch) [cos(pitch),  0, sin(pitch);
                   0,           1, 0;
                   -sin(pitch), 0, cos(pitch)];

Rroll = @(roll) [1, 0,         0;
                 0, cos(roll), -sin(roll);
                 0, sin(roll), cos(roll)];

% Hint use Ryaw,Rpitch,Rroll to find Q rather than writing the entire matrix
Q_BtoI = @(yaw,pitch,roll) Rroll(roll)*Rpitch(pitch)*Ryaw(yaw);
Q_Ito_B = @(yaw,pitch,roll) inv(Q_BtoI(yaw, pitch, roll));
```

Task 2:

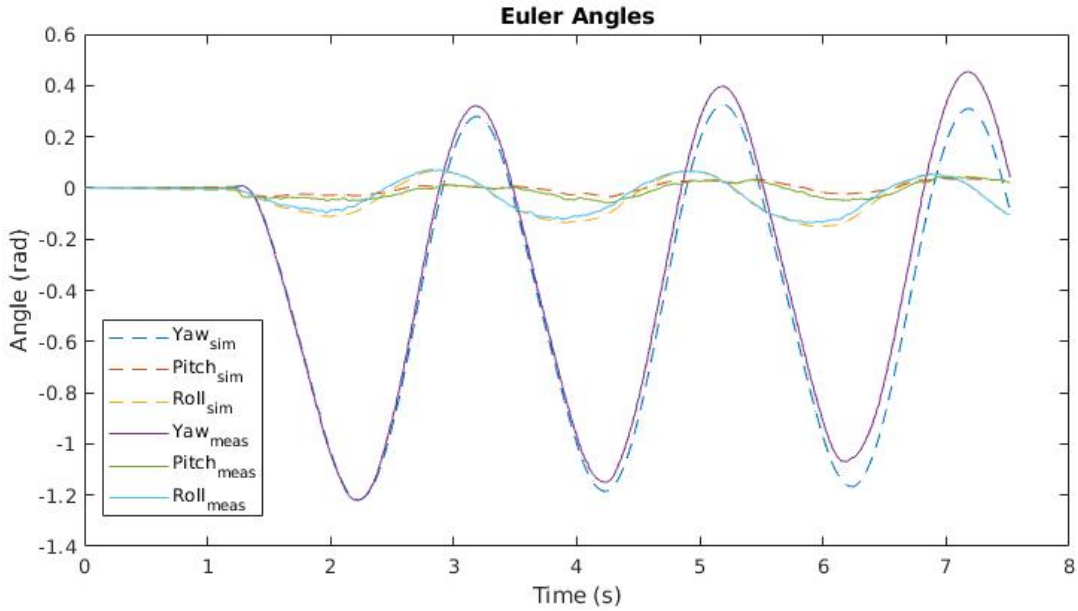
1. `odeEuler.m`

```
yaw = E(1);
pitch = E(2);
roll = E(3);

Winv =  (1/cos(pitch))*[0, sin(roll), cos(roll);
                        0, cos(roll)*cos(pitch), -sin(roll)*cos(pitch);
                        cos(pitch), sin(roll)*sin(pitch), cos(roll)*sin(pitch)];

angular_vel = [wx; wy; wz];

%dEuler is a 3x1 vector containing the derivatives of yaw,pitch,roll
dEuler = Winv*angular_vel;
```



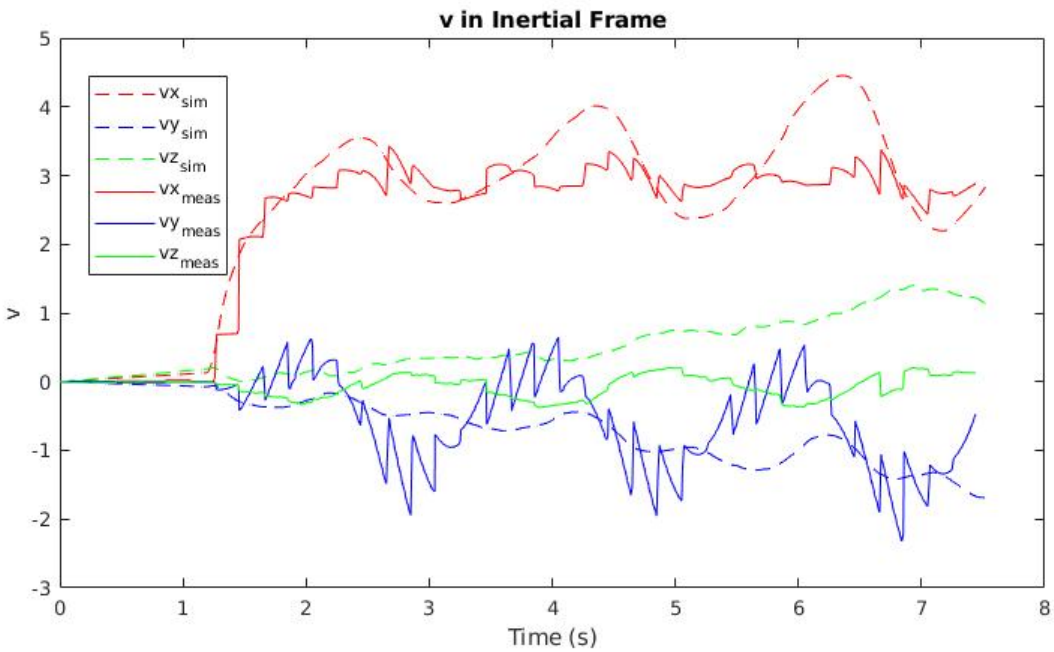
2. `odevB.m`

```
% v is a 3x1 vector [vx;vy;vz]
% function skew_sym(vector) will come in handy.

accel = [ax; ay; az - g];
angular_vel = [wx; wy; wz];
angular_skew_sym = skew_sym(angular_vel);

%% ****** Deliverable *****
%% Complete the ODEs for linear velocities in the body frame.
dvB = accel - angular_skew_sym*v;
```

3.



The measured velocity data is especially noisy because we are numerically differentiating already noisy position data from the GPS to obtain velocities. The simulation and measured velocities start to diverge after several seconds due to the fact that we are "dead reckoning" with the IMU data and integrating noise in the accelerometer will accumulate an error over time.

4. `odeID.m`

```
%% Complete the ODE for x-y-z coordinates in the inertial frame
%dIB is a 3x1 vector containing the derivatives of X, Y, Z

velocity = [vx; vy; vz];

dIB = Q_BtoI(yaw, pitch, roll)*velocity;
```