AERSP 424: NINT Example

First, let's review the gimbal equation – a good review of INS, and also useful preparation for things coming up. The lecture on numerical integration will also be a useful review if you're not current on the topic – please use Runge-Kutta 4th order to integrate. You will really need to use a computer to solve this – provide your code.

1. An aircraft is flying with a velocity of 60 knots in the direction of 'straight alone the aircraft's *x-axis*, which is defined by the vector from the cm straight out the nose'.
2. Its INS can sense that it is maneuvering in a constant condition: the roll-rate $p = 30\,\frac{\mathrm{deg}}{\mathrm{sec}}\left[\frac{\pi}{6}\frac{\mathrm{rad}}{\mathrm{sec}}\right]$, its angular velocity component $q = \cos\left(\frac{6}{\pi}t\right)$, and its angular velocity component $r = 3\sin\left(\frac{30}{\pi}t\right)$,
   - Using the gimbal equation, write the function that provides the rate of change of the aircraft Euler angles through time

```
% Gimbal equation
xdot = [
  1, tan(theta) * sin(phi), tan(theta) * cos(phi);
  0, cos(phi), -sin(phi);
  0, sin(phi) / cos(theta), cos(phi) / cos(theta)
  ] * [p; q; r];
```

   - Using the strapdown equation, write the function that updates the direction cosine matrix to convert vectors expressed in the body frame (in this case, velocity expressed in the body is [60knots, 0, 0] to velocity expressed in NED.

```
% Strapdown equation
UpdateMatrix = [0 -r q; r 0 -p; -q p 0];
Cdot = C * UpdateMatrix;
```

   - Integrate the state equations, and plot the time history of these signals for the time range 0 to 60 seconds:
     (1) $[p, q, r]$,
     (2) time rate of change of the Euler angles,
     (3) the Euler angles,
     (4) velocity expressed in NED, calculated using your updated DCM, and
     (5) position through time expressed in NED, where position at time 0 is [0,0,0].
       a) NOTE: You must update your DCM, but you do not need to plot it – demonstrate that it is correct by using it to transform your representation of velocity from 'body-axes' to 'North-East-Down'

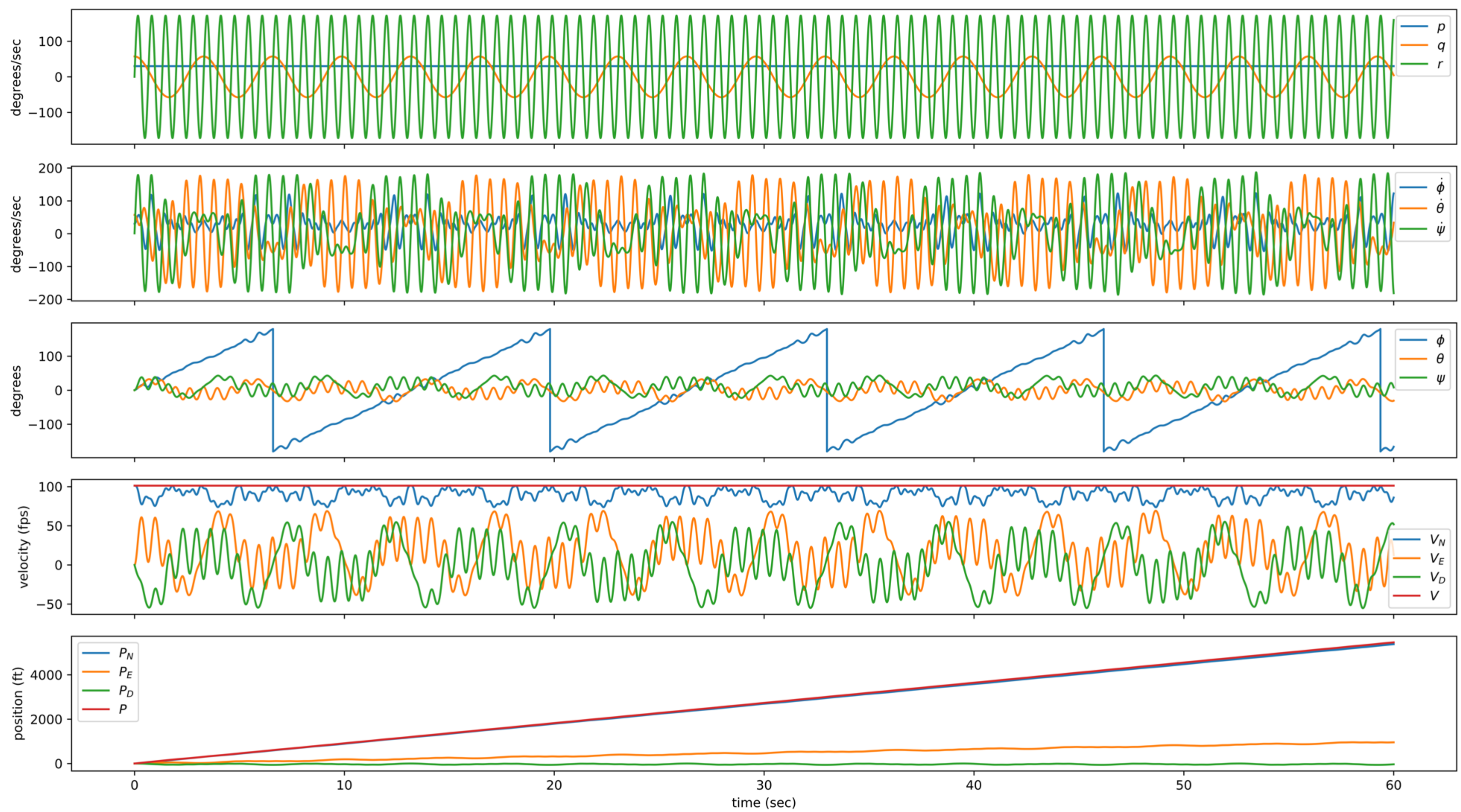
I've attached all of my MATLAB code for this problem, with a copy-paste of the relevant functions below, with figure outputs too! This is one crazy plane ride – a constant pitch rate is making it do loop-de-loops!

```
while time < endtime
    % calculate the 4 estimates of derivative over the time interval from t to t+dt
    [xdot1, Cdot1] = derivative(x(:, cntr), DCM, time, V_body);
    [xdot2, Cdot2] = derivative(x(:, cntr) + xdot1 * dt / 2, DCM + Cdot1 * dt / 2, time + dt / 2, V_body);
    [xdot3, Cdot3] = derivative(x(:, cntr) + xdot2 * dt / 2, DCM + Cdot2 * dt / 2, time + dt / 2, V_body);
    [xdot4, Cdot4] = derivative(x(:, cntr) + xdot3 * dt, DCM + Cdot3 * dt, time + dt, V_body);

    % calculate our 4th order Runge Kutta estimate of derivative
    totalxdot(:, cntr) = (xdot1 + 2 * xdot2 + 2 * xdot3 + xdot4) / 6;
    totalCdot = (Cdot1 + 2 * Cdot2 + 2 * Cdot3 + Cdot4) / 6;

    % update our state vector
    x(:, cntr + 1) = x(:, cntr) + totalxdot(:, cntr) * dt;
    DCM = DCM + totalCdot * dt;

    % do some bookkeeping
    cntr = cntr + 1; % increment our iteration counter
    time = time + dt; % increment time
    time_arr(cntr, 1) = time;
end
```


degrees/sec
p
q
r
degrees/sec
$\dot{\phi}$
$\dot{\theta}$
$\dot{\psi}$
degrees
ϕ
θ
ψ
velocity (fps)
V_N
V_E
V_D
V
position (ft)
P_N
P_E
P_D
P
time (sec)