Please also note the general rules for this HW:

1) Your HW report must be self-contained. Ensure all steps and values are included in the report, not just in your code. We won't run your code to verify your results; everything needed should be in the report itself.
2) Similar to HW 1: You will need to include your code with all the MATLAB formatting.

I. LOADING IN SENSOR DATA AND SOME ANALYSIS

You are provided with a CSV file that contains 4 pieces of information: the angular velocity components in the body frame and the corresponding timestamps for the measurements.

Use the following pseudo-code to extract the data in MATLAB.

```
clc
clear

T = readtable(Filename);

wx = T.wx;
wy = T.wy;
wz = T.wz;

% Step 1: Convert the time strings into datetime format
timeData = datetime(T.time, 'InputFormat', 'yyyy-MM-dd''T''HH:mm:ss.SSS''Z''',...
'TimeZone', 'UTC');

% Step 2: Calculate time differences from the first time in the list
timeDifferences = timeData - timeData(1);

% Step 3: Convert the differences to seconds
t = seconds(timeDifferences);
```

The initial Yaw-Pitch-Roll angles are given as:

$$\text{Yaw} = 30^0 \quad \text{Pitch} = 70^0 \quad \text{Roll} = 20^0$$

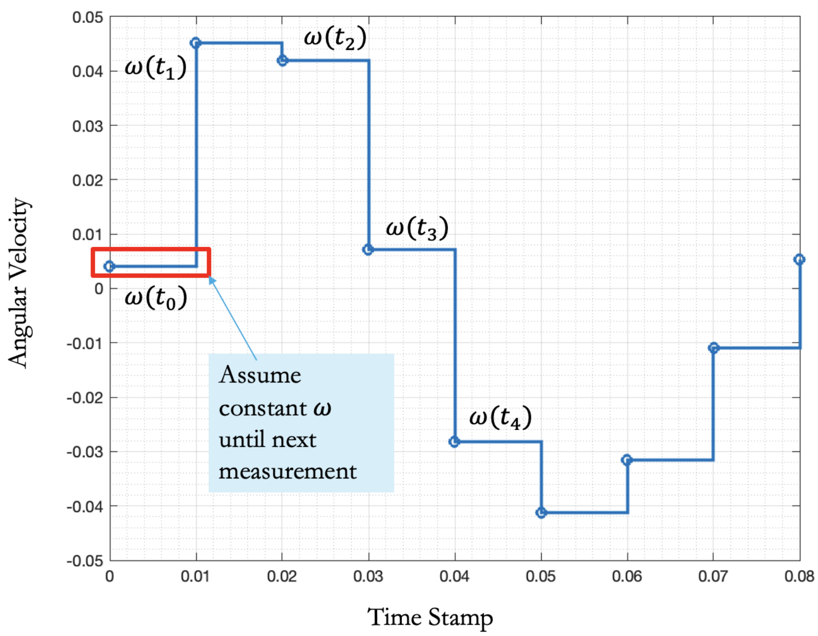
1) Which Euler-angle rotation sequence is this?
2) Convert the initial Yaw-Pitch-Roll values into a DCM.
3) Convert the initial Yaw-Pitch-Roll values into a quaternion.
4) Plot the angular velocities as a function of time.
5) Comment on the nature of the data: (a) what is the frequency of the sensor output. (b) Are the sensor outputs at equal time intervals?

II. PROPAGATION OF ATTITUDE MOTION USING DCM

1) **Numerically Propagate the DCM from time t_k to t_{k+1} ($k = 0, \cdots$) using the equations of motion derived in class.**

$$\dot{C}_{BN} = -[\tilde{\omega}]C_{BN}$$

You are to assume a zero-order-hold (ZOH) for the angular velocities. The zero-order hold (ZOH) is a method used in digital signal processing to reconstruct a continuous-time signal from a discrete-time signal. It assumes that each sample in a discrete-time signal remains constant (or "held") until the next sample arrives.
It is illustrated in the figure below:



2) **Analytically Propagate the DCM from time t_k to t_{k+1} ($k = 0, \cdots$) using the equations of motion derived in class.**

$$C_{BN}(t_{k+1}) = \text{expm}\left(-[\tilde{\omega}]\Delta t_k\right) \;\; C_{BN}(t_k)$$

where $\Delta t_k = t_{k+1} - t_k$

3) **Plot the time history of the error DCM.** The error is obtained by multiplying DCM obtained from numerical propagation with the inverse of DCM obtained from analytic propagation.

$$error(t_k) = C_{BN_{(numerical)}}(t_k)C^T_{BN_{(analytic)}}(t_k) - I_{3\times 3}$$

The quantity above must be zero if the error is zero.

4) **Plot the evolution of the Yaw-Pitch-Roll angles as a function of time.**

III. PROPAGATION OF ATTITUDE MOTION USING QUATERNION

1) **Analytically Propagate the Quaternions from time t_k to t_{k+1} using the equations of motion derived in class.**

$$\bar{\beta}(t_{k+1}) = \Phi(t_k, t_{k+1})\bar{\beta}(t_k), \quad \Phi(t_k, t_{k+1}) = e^{\frac{1}{2}B(\bar{\omega}_{\mathcal{B}/\mathcal{N}})\Delta t}, \quad B(\bar{\omega}_{\mathcal{B}/\mathcal{N}}) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}$$

2) **Plot the Quaternions and show that for all time the quaternion constraint is satisfied.**

3) **From the previous part, convert the roll-pitch-yaw angles obtained from analytical propagation and convert them to Quaternions.**

4) **Just like in the previous part, plot the time history of the error in the quaternion propagation.** The error is computed as:

$$error = \delta\beta - [1, 0, 0, 0]^T$$

where

$$\delta\beta = \beta_1 \otimes \beta_2^{-1}$$

where, β_1 is the analytically propagated quaternion, and β_2 is the (pitch-roll-yaw) from previous part converted to quaternions.