Table of Contents

PROBLEM 5

```
clear variables; close all; clc

data_table_acc = readtable('Accelerometer_Att_EKF');
data_table_gyro = readtable('Gyroscope_att_EKF');
data_table_mag = readtable('Magnetometer_Att_EKF');

clc

% CORRECT VALUES FOR THESE BIASES AND VARIANCES AS COMPUTED IN
 EXPERIMENT 1
bias_acc = [1.948244;
               1.926003;
               -3.76083]; % biases in accelerometer x,y,z


bias_mag = [-12.11214937;
               -19.67616054;
               22.73696197]; % biases in magnetometer x,y,z

bias_gyro = [0.00001194560806;
               -0.00000912316961;
               -0.00000169621783]; % biases in gyro x,y,z
var_acc  = [4.94; 5.23; 13.5]*10^-5;
var_gyro = [2.1; 2.6; 4.9]*10^-6;
var_mag     = [0.467657;0.744017;0.463155];
```

```
Warning: Column headers from the file were
modified to make them valid MATLAB
identifiers before creating variable names
for the table. The original column headers
are saved in the VariableDescriptions
property.
Set 'PreserveVariableNames' to true to use
the original column headers as table
variable names.
Warning: Column headers from the file were
modified to make them valid MATLAB
identifiers before creating variable names
for the table. The original column headers
```

```
are saved in the VariableDescriptions
property.
Set 'PreserveVariableNames' to true to use
the original column headers as table
variable names.
Warning: Column headers from the file were
modified to make them valid MATLAB
identifiers before creating variable names
for the table. The original column headers
are saved in the VariableDescriptions
property.
Set 'PreserveVariableNames' to true to use
the original column headers as table
variable names.
```

Local gravitational acceleration

```
g = 9.80333; % m/s/s
```

Accelerometers

```
acc_x = data_table_acc{:, 2};
acc_y = data_table_acc{:, 3};
acc_z = data_table_acc{:, 4};
```

Rate gyros

```
gyro_x = data_table_gyro{:, 2};
gyro_y = data_table_gyro{:, 3};
gyro_z = data_table_gyro{:, 4};
```

Magnetometers

```
mag_xb = data_table_mag{:, 2};
mag_yb = data_table_mag{:, 3};
mag_zb = data_table_mag{:, 4};
```

Remove bias

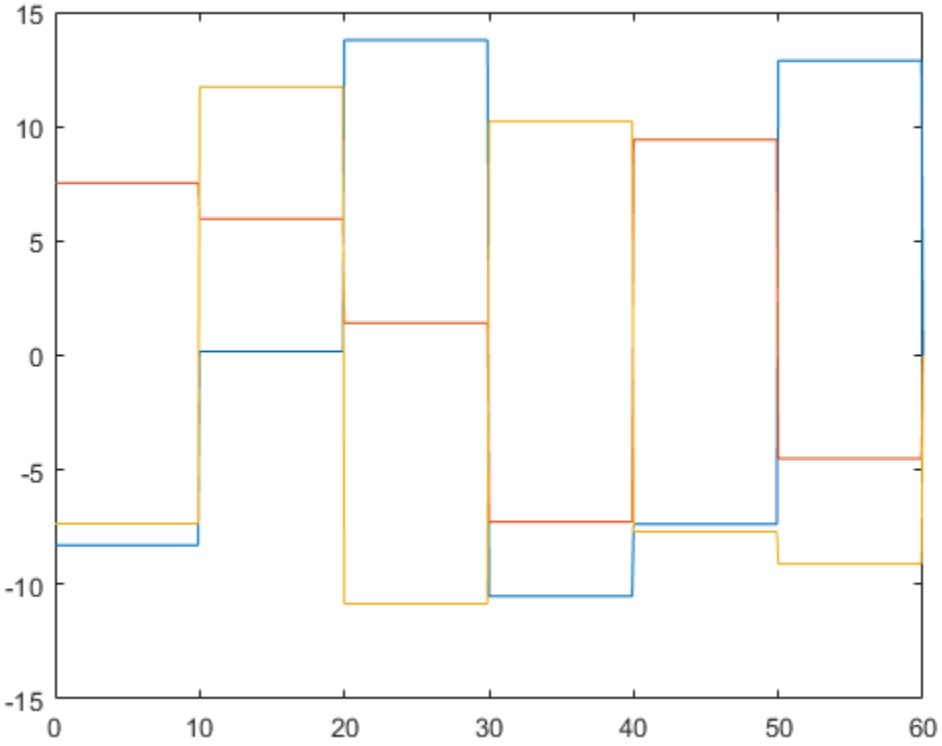
```
acc_x_wo_bias = acc_x - bias_acc(1);
acc_y_wo_bias = acc_y - bias_acc(2);
acc_z_wo_bias = acc_z - bias_acc(3);

mag_xb_wo_bias = mag_xb - bias_mag(1);
mag_yb_wo_bias = mag_yb - bias_mag(2);
mag_zb_wo_bias = mag_zb - bias_mag(3);
```

Pitch and Roll

```
roll_data = atan( acc_y_wo_bias ./ acc_z_wo_bias );
```

```
pitch_data = asin( acc_x_wo_bias / g );

roll_mu  = mean( roll_data );
pitch_mu = mean( pitch_data );

roll_var = var(roll_data);
pitch_var = var(pitch_data);


time_stamps_mag = data_table_mag{1:20, 1};
mag_x = data_table_mag{1:20, 2};
mag_y = data_table_mag{1:20, 3};
mag_z = data_table_mag{1:20, 4};

time_stamps = 0:0.1:60;
n_t  = numel(time_stamps);
omega_b = zeros(3, n_t);

n_pieces = 6;
for m1 = 0:(n_pieces-1)
 omega_b(:, (m1*100 + 1):((m1+1)*100)) = kron( ...
  ((-15 + 30*rand(3,1))*pi/180), ones(1, 100));
end

plot(time_stamps, omega_b*180/pi)

data_table_sheet1 = table( ...
 time_stamps', omega_b(1,: )', omega_b(2,: )',
 omega_b(3,: )', 'VariableNames', ...
 {'Time (s)'; 'Gyroscope x (rad/s)'; 'Gyroscope y (rad/s)'; 'Gyroscope
 z (rad/s)'});

data_table_sheet2 = table( ...
 time_stamps_mag, mag_x, mag_y, mag_z, 'VariableNames', ...
 {'Time (s)'; 'Magnetic field x (mu T)'; 'Magnetic field y (mu
 T)'; 'Magnetic field z (mu T)'});

writetable(data_table_sheet1, 'data_exp4_sample.xls', 'Sheet', 1)
writetable(data_table_sheet2, 'data_exp4_sample.xls', 'Sheet', 2)


function euler_dot = euler321_kinematics( t, euler_angles, omega_b_ )

 phi   = euler_angles(1);
 theta = euler_angles(2);

 Hinv_321 = (1 / cos(theta) ) * [cos(theta), sin(phi)*sin(theta),
 cos(phi)*sin(theta); ...
  0, cos(phi)*cos(theta), -sin(phi)*cos(theta);
  0, sin(phi), cos(phi)];

 euler_dot = Hinv_321 * rate_gyro_readings;
```

```
end
```



Published with MATLAB® R2020a