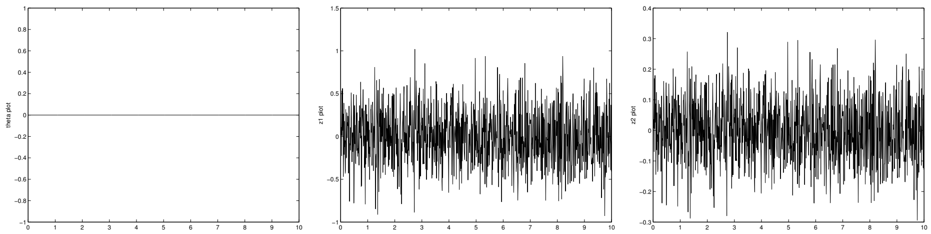
One dimensional attitude estimation

In this problem, we consider one-dimensional attitude estimation problem: estimation of θ. Two sensors are used to estimate θ: an inclinometer (z_1) and a gyroscope (z_2).

$$\begin{aligned} z_1(t) &= \theta(t) + v_1(t) \\ z_2(t) &= \dot{\theta}(t) + b_g + v_2(t), \end{aligned}$$

where v_1 and v_2 are measurement noises. In the first experiment, attitude is measured while maintaining $\theta(t) = 0$ and $b_g = 0$.



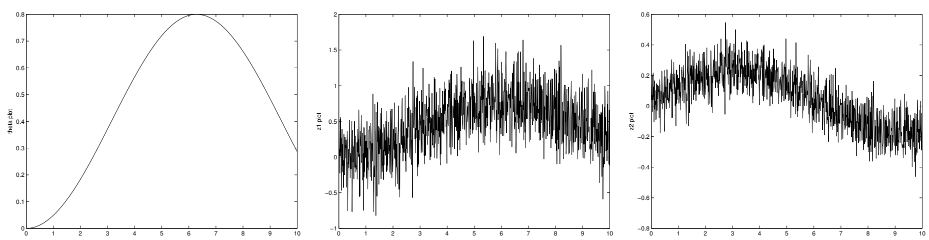
(a) θ (b) $z_1(t)$ (c) $z_2(t)$

You can download the above data from from uclass (`attitude1.mat`) and type the following in the matlab:

```
load attitude1.mat
```

You can see there are four variables: t, z1, z2, theta. Note that t is time profile of the three data z1, z2, theta. For example, the first graph is plotted using `plot(t,theta)`.

In the second experiment, θ is varying. The above data is in `attitude2.mat`.



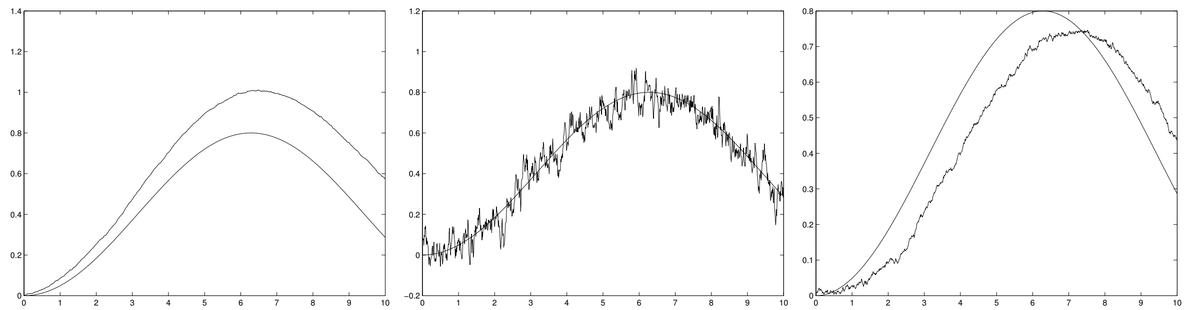
(a) θ (b) $z_1(t)$ (c) $z_2(t)$

Attitude estimation using gyroscope integration

```
N = max(size(z1));
thetahat = zeros(N,1);
for i = 2:N
    thetahat(i) = thetahat(i-1) + (t(i)-t(i-1))*z2(i-1);
end
plot(t,theta,'r',t,thetahat,'b');
```

Attitude estimation using z_1

```
thetahat2 = zeros(N,1);
for i = 2:N
    thetahat2(i) = a * thetahat2(i-1) + (1-a) * z1(i);
end
plot(t,theta,'r',t,thetahat2,'b');
```



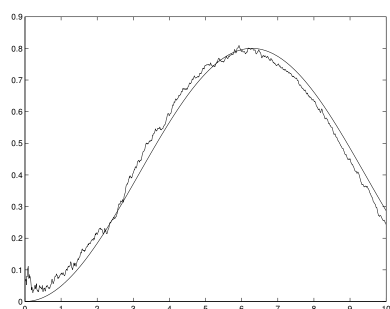
(a) gyroscope integration (b) inclinometer + low pass filter ($a = 0.9$) (c) $a = 0.99$

The following problems are for the second experiment.

(1) Find $\hat{\theta}$ using the complementary filter. Try to choose optimal a value

(2) Find $\hat{\theta}$ using the indirect Kalman filter considering the gyroscope bias. Can you guess the gyroscope bias?

For all problems, compute $\sum(\theta - \hat{\theta})^2$. You should explain your design process and submit m-files.



Complementary filter code

```
load attitude2.mat



N = length(z1);
thetahat = zeros(N,1);
thetahat(1) = z1(1);
a = 2;
for i = 2:N
    T = t(i) - t(i-1);
    thetahat(i) = (1 - a * T) * thetahat(i-1) + T * a * z1(i) + T * z2(i);
end

plot(t,theta,'r',t,thetahat,'b')
norm( theta - thetahat)^2
```

Indirect Kalman filter without bias term

```
load attitude1.mat
r = var(z1);
q = var(z2);

load attitude2.mat

N = length(z1);
thetahat = zeros(N,1);

P = r;
thetahat(1) = z1(1);
% numerical integration error
alpha = 2;
for i = 2:N
    T = t(i) - t(i-1);
    thetahat(i) = thetahat(i-1) + T * z2(i-1);
    P = P + alpha * q * T^2;
    K = P * inv(P + r);
    delta_theta = K * (z1(i) - thetahat(i));
    thetahat(i) = thetahat(i) + delta_theta;
    P = (1 - K) * P;
end

norm( theta - thetahat)^2
plot(t,theta,'r',t,thetahat,'b')
```

Indirect Kalman filter a bias term

```
load attitude1.mat
r = var(z1);
q = var(z2);

load attitude2.mat

alpha = 2;
N = length(z1);
thetahat = zeros(N,1);
thetahat(1) = z1(1);
bghat = 0;
bghat_history = zeros(1,N);
P = ???;   % initial P
H = [ 1  0 ];
for i = 2:N
    T = t(i) - t(i-1);
    thetahat(i) = thetahat(i-1) + T * (z2(i-1) - bghat);

    % time update
    P = ???;
    K = P * H' * inv(H * P * H' + r);

    % measurement update
    delta_hat = K * (z1(i) - thetahat(i));

    thetahat(i) = thetahat(i) + ???;
    bghat = bghat + ???;

    P = (eye(2) - K * H) * P;
    P = 0.5 * (P + P');
   bghat_history(i) = bghat;
 end
```