Homework 2

1. Two random variables x_1 and x_2 have a joint PDF that is uniform inside the circle (in the x_1-x_2 plane) with radius 2, and zero outside the circle.

(a) Find the math expression of the joint PDF function.
(b) Find the conditional PDF $P_{x_2|x_1}(x_2|x_1 = 0.5)$?
(c) Are the two random variables uncorrelated?
(d) Are the two random variables statistically independent? (Hint: find $p_x(x_1)$ and $p_x(x_2)$ and check if $p_{x_1x_2}(x_1, x_2) = p_{x_1}(x_1)p_{x_2}(x_2)$)

Solution:
Assuming $x_1 = x$ and $x_2 = y$. For a uniform distribution inside a circle, the joint PDF is given by:

$$f_{xy}(x, y) = \begin{cases} c & x^2 + y^2 \leq r^2 \\ 0 & x^2 + y^2 > r^2 \end{cases}$$

To find c, the double integral of the PDF where c is valid must be taken.

$$1 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{xy}(x, y) \; dxdy$$
$$1 = \int \int_{x^2+y^2 \leq r^2} c \; dxdy$$

Where this double integral is equivalent to the area of a circle with radius r scaled by c, which completes the joint PDF above:

$$1 = \pi r^2 c = \pi 2^2 c$$
$$c = \frac{1}{4\pi}$$

To evaluate the PDF at $x = 0.5$, the marginal density functions of x and y must first be calculated.

$$f_x(x) = \int_{-\infty}^{\infty} f_{xy}(x, y) dy$$
$$f_x(x) = \int_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}} \frac{1}{4\pi} dy = \frac{1}{4\pi} y \Big|_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}} = \frac{1}{4\pi}(\sqrt{4 - x^2} - (-\sqrt{4 - x^2}))$$
$$f_x(x) = \frac{2\sqrt{4 - x^2}}{4\pi}$$

And by symmetry between the axes:

$$f_y(y) = \frac{2\sqrt{4-y^2}}{4\pi}$$

Evaluating the joint PDF at $x = 0.5$:

$$\begin{aligned} f_{y|x}(y|x=0.5) &= \frac{f_{xy}(0.5,y)}{f_x(0.5)} \\ f_{y|x}(y|x=0.5) &= \frac{\frac{1}{4\pi}}{\frac{2\sqrt{4-0.5^2}}{4\pi}} \\ f_{y|x}(y|x=0.5) &= \frac{1}{2\sqrt{3.75}} \end{aligned}$$

To determine if the random variables are uncorrelated, the following definition can be used:

$$\begin{aligned} E\{xy\} = E\{x\}E\{y\} = \bar{x}\bar{y} &= 0 \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf_{xy}(x,y)\ dxdy \\ &= \frac{1}{4\pi}\int_{-\sqrt{4-y^2}}^{\sqrt{4-y^2}}\int_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}} xy\ dxdy \\ &= \frac{1}{4\pi}\left(\frac{y}{2}\Big|_{-\sqrt{4-y^2}}^{\sqrt{4-y^2}}\right)\left(\frac{x}{2}\Big|_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}}\right) \\ &= \frac{1}{4\pi}(0)(0) \\ &= 0 \end{aligned}$$

Therefore the two variables are uncorrelated. Last, to determine if the variables are independent, a separate rule can be used.

$$\begin{aligned} f_xy(x,y) &= f_xf_y \\ \frac{1}{4\pi} &= \frac{2\sqrt{4-x^2}}{4\pi}\frac{2\sqrt{4-y^2}}{4\pi} \\ \frac{1}{4\pi} &\neq \frac{1}{2\pi}\sqrt{4-x^2}\sqrt{4-y^2} \end{aligned}$$

Therefore, the variables are not independent.

2. The stationary process $x(t)$ has an autocorrelation function of the form:

$$R_x(\tau) = \sigma^2 e^{-\beta|\tau|}$$

Another process $y(t)$ is related to $x(t)$ by the deterministic equation:

$$y(t) = ax(t) + b$$

where the constants a and b are known.

(a) What it the autocorrelation function for y(t) ?
(b) What is the crosscorrelation function $R_{xy}(\tau) = E\{x(t)y(t+\tau)\}$?

Solution:
Applying the definition of an autocorrelation function:

$$\begin{aligned}
R_x(\tau) &= E\{x(t)x(t+\tau)\} = \sigma^2 e^{-\beta|\tau|} \\
R_y(\tau) &= E\{x(t)x(t+\tau)\} \\
&= E\{[ax(t)+b][ax(t+\tau)+b]\} \\
&= E\{b^2 + a^2 x(t)x(t+\tau) + abx(t) + abx(t+\tau)\} \\
&= b^2 + a^2\left(\sigma^2 e^{-\beta|\tau|}\right) + 2ab\mu
\end{aligned}$$

Where μ is the expected value (or mean) of x. To find the crosscorrelation the function defined above is used:

$$\begin{aligned}
R_{xy}(\tau) &= E\{x(t)y(t+\tau)\} \\
&= E\{x(t)[ax(t+\tau)+b]\} \\
&= E\{bx(t) + ax(t)x(t+\tau)\} \\
&= b\mu + a\left(\sigma^2 e^{-beta|\tau|}\right)
\end{aligned}$$

3. Use least squares to identify a gyroscopes scale factor (a) and bias (b). Simulate the gyroscope using:

$$\begin{aligned} g(k) &= ar(k) + b + n(k) \\ n &\sim N(0, \sigma = 0.3\text{deg/s}) \\ r(k) &= 100\sin(\omega t) \end{aligned}$$

(a) perform the least squares with 10 samples (make sure to pick ω so that you get one full cycle in 10 samples.

(b) Repeat part a 1000 times and calculate the mean and standard deviation of the estimate errors (this is known as a Monte Carlo Simulation). Compare the results to the theoretically expected mean and standard deviation.

(c) Repeat part (a) and (b) using 1000 samples. What does the theoretical and Monte Carlo standard deviation of the estimated errors approach?

(d) Set up the problem to run as a recursive least squares and plot the coefficients and theoretical standard deviation of the estimate error and the actual estimate error as a function of time.

Solution:

To solve a least squares problem, the geometry matrix and measurement vector must be defined.

$$\begin{aligned} H &= \begin{bmatrix} 100\sin(\omega t) & 1 \end{bmatrix} \\ y &= a(100\sin(\omega t)) + b + n \\ x &= (H^T H)^{-1} H^T y \end{aligned}$$

The H and y matrices would scale column wise to match the number of time points sampled and x is the state vector containing a and b. This along with the definitions of a, b, and ω were input in MATLAB as follows:

```
% angular frequency, scale factor, bias
w = rad2deg(2*pi*1); a = 0.5;      b = 3;
g = @(t) a*(100*sind(w.*t)) + b + 0.3.*randn(size(t)); % gyro measurement
G = @(t) [100*sind(w.*t), ones(size(t))];              % geometry matrix
```

For part a, and as an example of how all the following least squares solutions would be solved:

```
% PART A
dt = 0.1; % gyro update rate (1/batch_size)
t = (dt:dt:1)';
y = g(t);
H = G(t);
x = (H'*H)\(H'*y);
```

The least squares solution in part a is $x = \begin{bmatrix} 0.501528 & 3.112882 \end{bmatrix}^T$ (where a is $x(1)$ and b is $x(2)$). Repeating this method, with a batch size of 10, 1000 times for the monte carlo simulation:

$$\sigma_{estimates} = \begin{bmatrix} 0.001348 & 0.097362 \end{bmatrix}^T$$
$$\mu_{estimates} = \begin{bmatrix} 0.499989 & 3.002229 \end{bmatrix}^T$$
$$\sigma_{theoretical} = \begin{bmatrix} 0.001342 & 0.094868 \end{bmatrix}^T$$
$$\mu_{theoretical} = \begin{bmatrix} 0.500000 & 3.000000 \end{bmatrix}^T$$

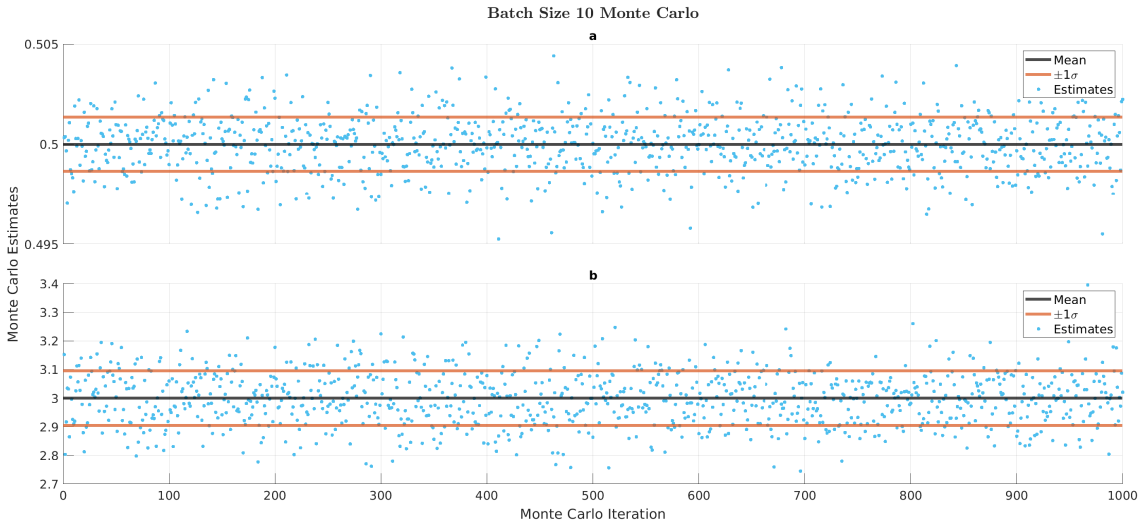

Figure 1: Monte Carlo with batch size of 10.

Where the theoretical standard deviation values were calculated from the covariance matrix as follows:

$$P = \sigma^2 (H^T H)^{-1} = 0.3^2 (H^T H)^{-1}$$
$$\sigma_{theoretical} = \begin{bmatrix} \sqrt{P(1,1)} & \sqrt{P(2,2)} \end{bmatrix}^T$$

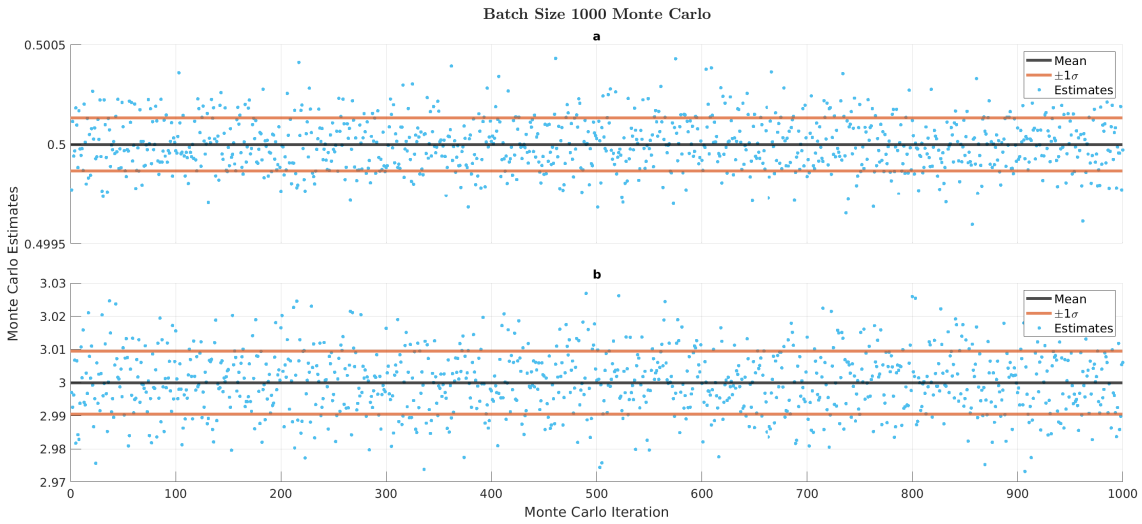

Figure 2: Monte Carlo with batch size of 1000.

Looking at the above values, the estimates are extremely close the the expected values, with all of the values for a being closer together than the values for b. Repeating the monte carlo simulation with a batch size of 1000:

$$\sigma_{estimates} = \begin{bmatrix} 0.000131 & 0.009683 \end{bmatrix}^T$$
$$\mu_{estimates} = \begin{bmatrix} 0.500000 & 3.000223 \end{bmatrix}^T$$
$$\sigma_{theoretical} = \begin{bmatrix} 0.000134 & 0.009487 \end{bmatrix}^T$$
$$\mu_{theoretical} = \begin{bmatrix} 0.500000 & 3.000000 \end{bmatrix}^T$$

Comparing these estimates to the estimates of a batch size of 10, it is first notable that the estimate mean error decreases as well as the standard deviations converging towards 0. Both decrease by approximately a factor of 10 (notice the scale on each plot).
Converting the problem into a recursive least squares setup instead of batched least squares required the following equation adjustments:

$$Q_k = Q_{k-1} + H_k^T H_k$$
$$\partial x = Q_k^{-1} H_k^T \left(y_k - H_k x_{k-1}\right)$$
$$x_k = x_{k-1} + \partial x$$

This was implemented in MATLAB with the following code for 100 iterations between 0 and 1 second:

```
dt = 1/100;
t = (dt:dt:1)';
```

```
R = (0.3^2 * eye(1))^-1;
x = zeros(2,100);
P = zeros(2,2,100);

for i = 1:100
    if i == 1
        y = g(t(1));
        H = G(t(1));
        Q = H'*H;
        x(:,1) = Q^-1 * (H'*y);
        P(:,:,1) = 0.3^2 * Q^-1;
    else
        y = g(t(i));
        H = G(t(i));
        Q = Q + H'*H;
        dx = Q^-1 * H' * (y - H*x(:,i-1));

        x(:,i) = x(:,i-1) + dx;
        P(:,:,i) = 0.3^2 * Q^-1;
    end
end
```

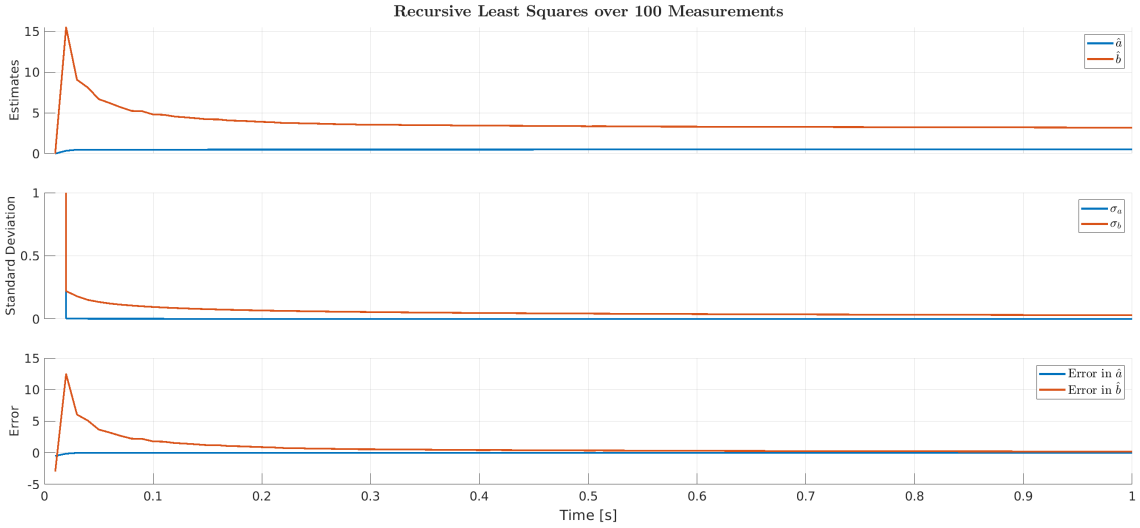

Figure 3: Recursive Least Squares over Time.

As shown, the first iteration results in a poor solution with a huge standard deviation on the scale of 10^7 (the axis was scaled to between 0 and 1). After the second iteration, the standard deviation reduces to more reasonable values, and by the end of the run, the estimates have converged to approximately the correct values.

4. Least Squares for System I.D. Simulate the following discrete system with a normal random input and output noise:

$$G(z) = \frac{0.25(z - 0.8)}{z^2 - 1.90z + 0.95}$$

(a) Develop the H matrix for the least squares solution.

(b) Use least squares to estimate the coefficients of the above Transfer Function. How good is the fit? Plot the bode response of the I.D. TF and the simulated TF on the same plot. How much relative noise has been added (SNR - signal to noise ratio), plot y and Y on the same plot.

(c) Repeat the estimation process about 10 times using new values for the noise vector each time. Compute the mean and standard deviation of your parameter estimates. Compare the computed values of the parameter statistics with those predicted by the theory based on the known value of the noise statistics.

(d) Now use sigma between 0.1 and 1.0 and repeat parts b and c.

(e) What can you conclude about using least squares for system id with large amounts of noise?

Solution:

To develop a geometry matrix for the system, the transfer function is generalized to the following:

$$\frac{Y}{U} = \frac{az + b}{z^2 + cz + d}$$

Expanding and performing an inverse z-transform with U defined as the input and Y defined as the output:

$$U(az + b) = Y(z^2 + cz + d)$$
$$aU_{k+1} + bU_k = Y_{k+2} + cY_{k+1} + dY_k$$

Taking Y_{k+2} as the measurements of the system, the geometry matrix can be defined as follows:

$$H = \begin{bmatrix} U_{k+1} & U_k & -Y_{k+1} & -Y_k \end{bmatrix}$$

And the full system.

$$Y_{k+2} = \begin{bmatrix} U_{k+1} & U_k & -Y_{k+1} & -Y_k \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Using the following MATLAB code to estimate the coefficients:

```
% discrete system
num = 0.25 .* [1, -0.8];
```

```
den = [1, -1.9, 0.95];
G = tf(num,den, 1);

% simulated discrete system
U = randn(1000,1);
y = dlsim(num,den,U);
sigma = 0.01;
Y = y + sigma*randn(1000,1);

% least squares coefficient estimation
H = [U(2:end-1), U(1:end-2), -Y(2:end-1), -Y(1:end-2)];
m = Y(3:end);
x = (H'*H)^-1*H'*m;
P = sigma^2 * (H'*H)^-1;

% recreated system
numid = [x(1), x(2)];
denid = [1, x(3), x(4)];
Gid = tf(numid,denid, 1);
SNR = std(Y) / sigma;
```

Resulting in the following bode plot and signal plot.

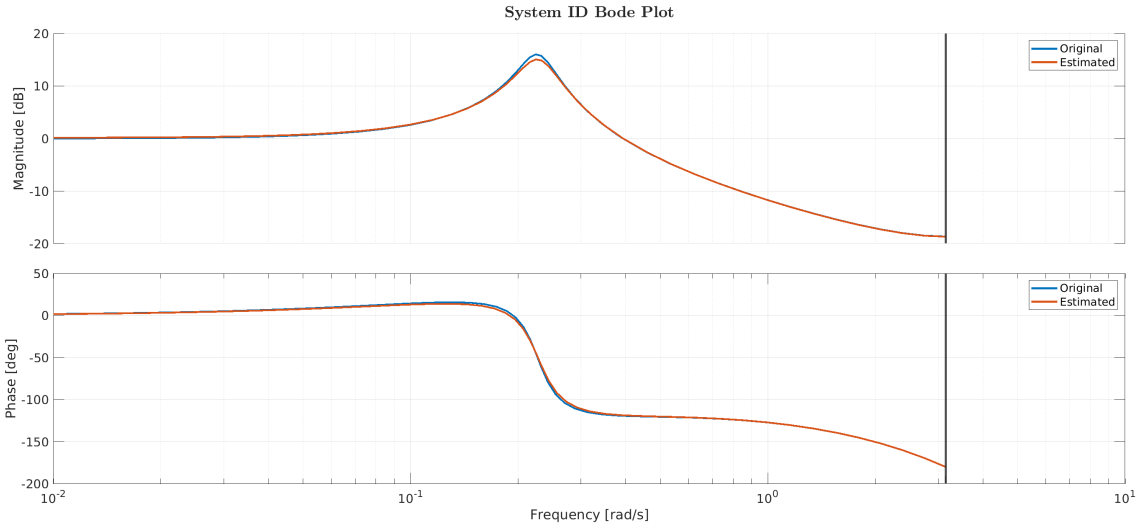

Figure 4: Bode Plot of Simulated and I.D. System.

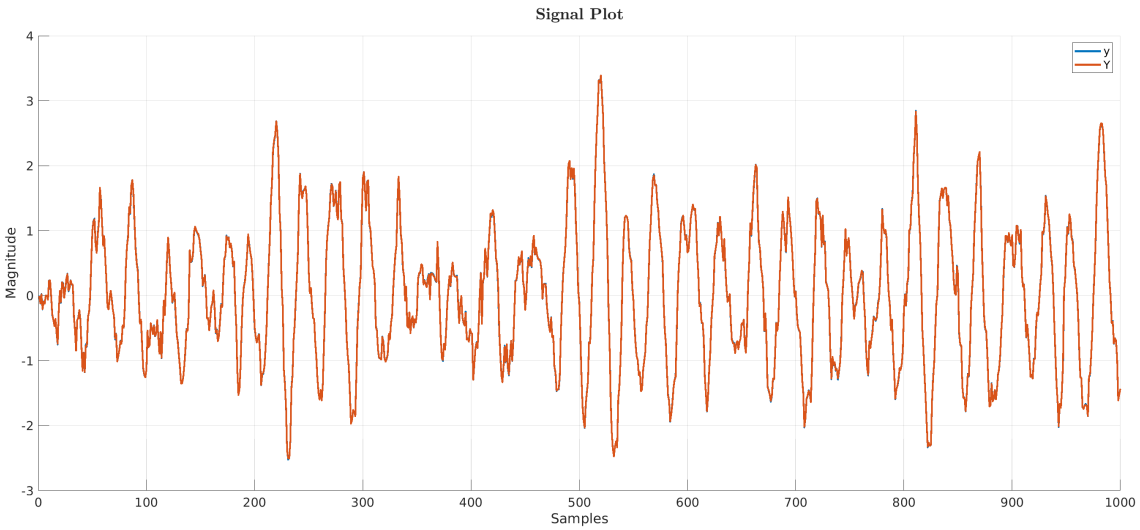

Figure 5: Simulated and Recreated Signal.

As shown, the recreated system is almost exactly the same as the original due to the small amount of noise added to the signal. The calculated estimates and signal to noise ratio were:

$$x = \begin{bmatrix} 0.251377 & -0.199053 & -1.892648 & 0.942921 \end{bmatrix}^T$$
$$SNR = 95.299670$$

Running a 10 iteration monte carlo to get an idea of the system statistics:

$$\sigma_{estimated} = \begin{bmatrix} 0.000817 & 0.001139 & 0.001296 & 0.001205 \end{bmatrix}^T$$
$$\mu_{estimated} = \begin{bmatrix} 0.249995 & -0.197918 & -1.892480 & 0.942775 \end{bmatrix}^T$$
$$\sigma_{theoretical} = \begin{bmatrix} 0.000315 & 0.000512 & 0.001603 & 0.001554 \end{bmatrix}^T$$
$$\mu_{theoretical} = \begin{bmatrix} 0.250000 & -0.200000 & -1.900000 & 0.950000 \end{bmatrix}^T$$

While the means are almost equivalent, the standard deviations are not as close likely due to the small sample size used in the monte carlo simulation. Iterating this monte carlo simulation through a loop of σ values between 0.1 and 1 results in the following characteristics:

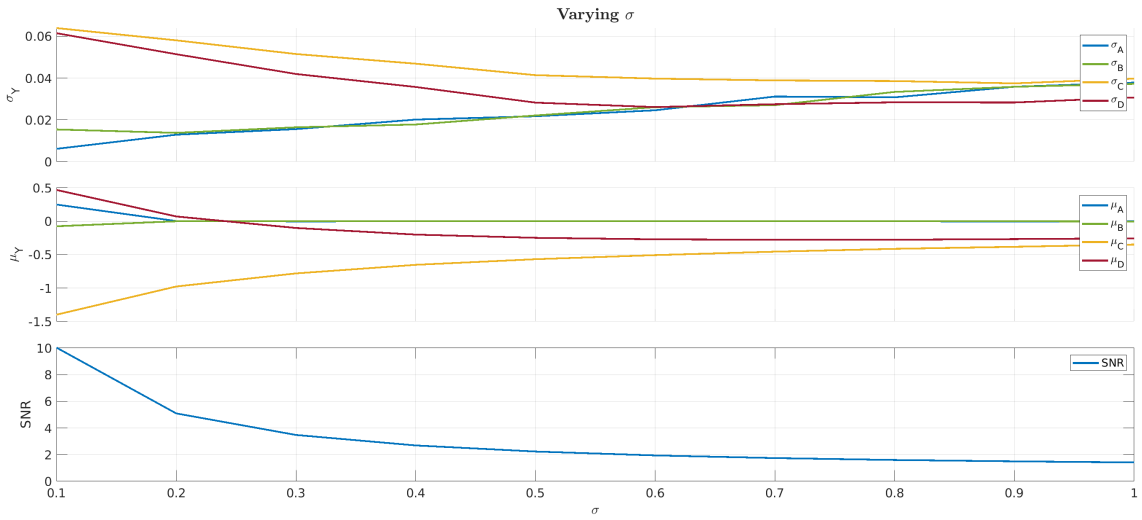

Figure 6: Effects of Higher Noise Levels.

As displayed, the mean estimates drift far from the actual values and the signal to noise ratio gets extremely low as the recreated signal becomes dominated by the noise. Redoing the bode plot and signal plot for a standard deviation of 1.

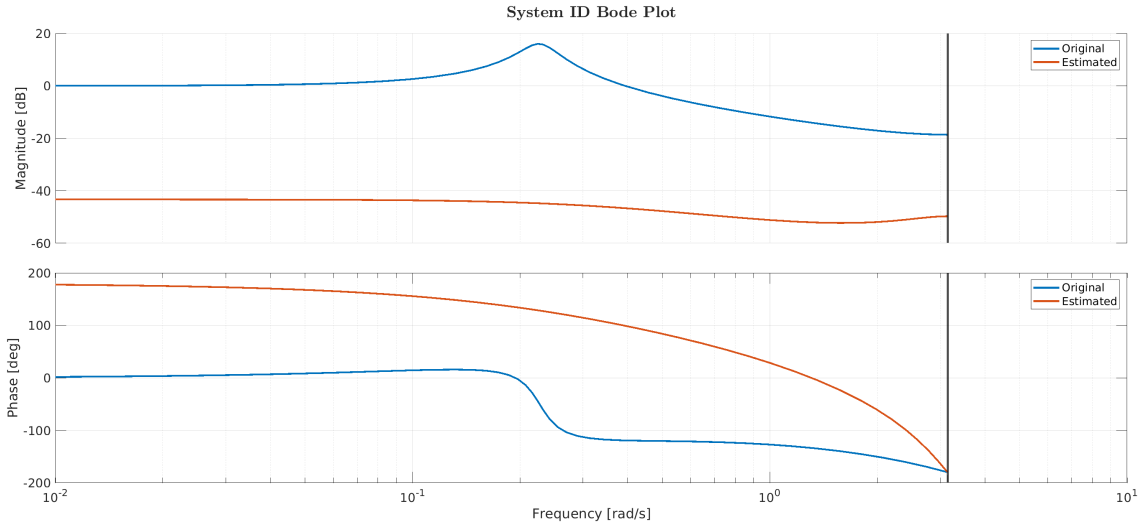

Figure 7: Bode Plot of Simulated and I.D. System with $\sigma = 1$.

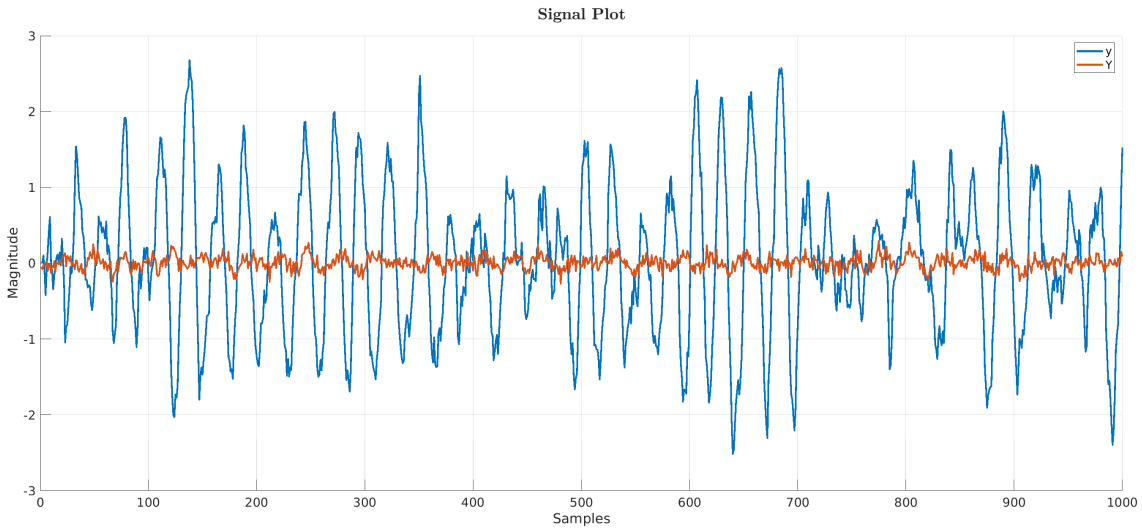

Figure 8: Simulated and Recreated Signal with $\sigma = 1$.

As shown the input signal and output signal are completely uncorrelated. This leads to the conclusion that least squares system identification with high noise levels is not plausible as the noise overpowers the fit of the signal.

5. Justification of white noise for certain problems. Consider two problems:

$$S_y(j\omega) = |G(j\omega)|^2 S_\omega(j\omega)$$

(i) Simple first order low-pass filter with bandlimited white noise as the input:

$$y = G(s)\omega \quad \ni \quad S_y(j\omega) = |G(j\omega)|^2 S_\omega(j\omega)$$

$$S_1(\omega) = \begin{cases} A & |\omega| \leq \omega_c \\ 0 & |\omega| > \omega_c \end{cases}$$

$$G(s) = \frac{1}{T_\omega s + 1}$$

(ii) The same low pass system, but with pure white noise as the input.

$$S_2(\omega) = A \ \forall \ \omega$$

$$G(s) = \frac{1}{T_\omega s + 1}$$

The first case seems quite plausible, but the second case has an input with infinite variance and so is not physically realizable. However, the white noise assumption simplifies the system analysis

significantly, so it is important to see if the assumption is justified. We test this with our two examples above:

(a) Sketch the noise PSD and $|G(j\omega)|$ for a reasonable value of T_ω and ω_c to compare the two cases.
(b) Determine the $S_y(j\omega)$ for the two cases. Sketch these too.
(c) Determine $E\{y^2\}$ for the two cases.
(d) Use these results to justify the following statement: "If the input spectrum is flat considerably beyond the system bandwidth, there is little error introduced by assuming that the input spectrum is flat out to infinity."

Solution:

Sketching the plots for of the input PSD of band-limited white noise, pure white noise, and the bode plot of the transfer function.

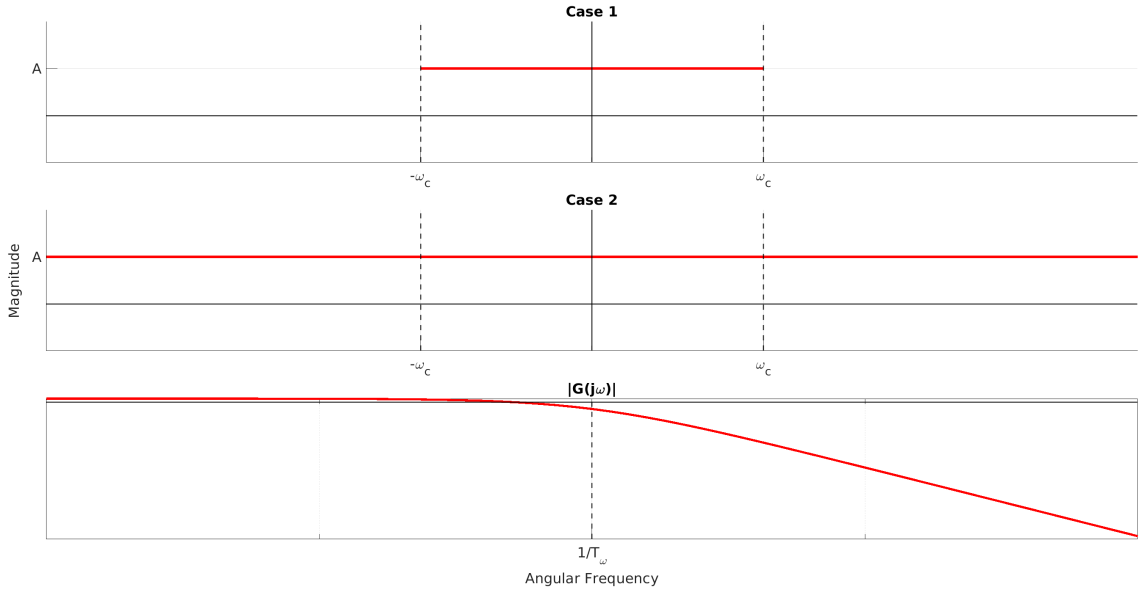

Figure 9: Drawings of the Input PSDs.

To determine the PSD of the output using the given equation:

$$\begin{aligned} S_y(j\omega) &= |G(j\omega)|^2 S_\omega(j\omega) \\ &= \left| \frac{1}{T_\omega(j\omega) + 1} \right|^2 S_{1|2}(\omega) \\ &= \frac{1}{T_\omega^2 \omega^2 + 1} S_{1|2}(\omega) \end{aligned}$$

For case 1, $S_{1|2} = A$ between $-\omega_c \leq \omega \leq \omega_c$ and for case 2, $S_{1|2} = A$ is always A. Sketching the two output PSDs.

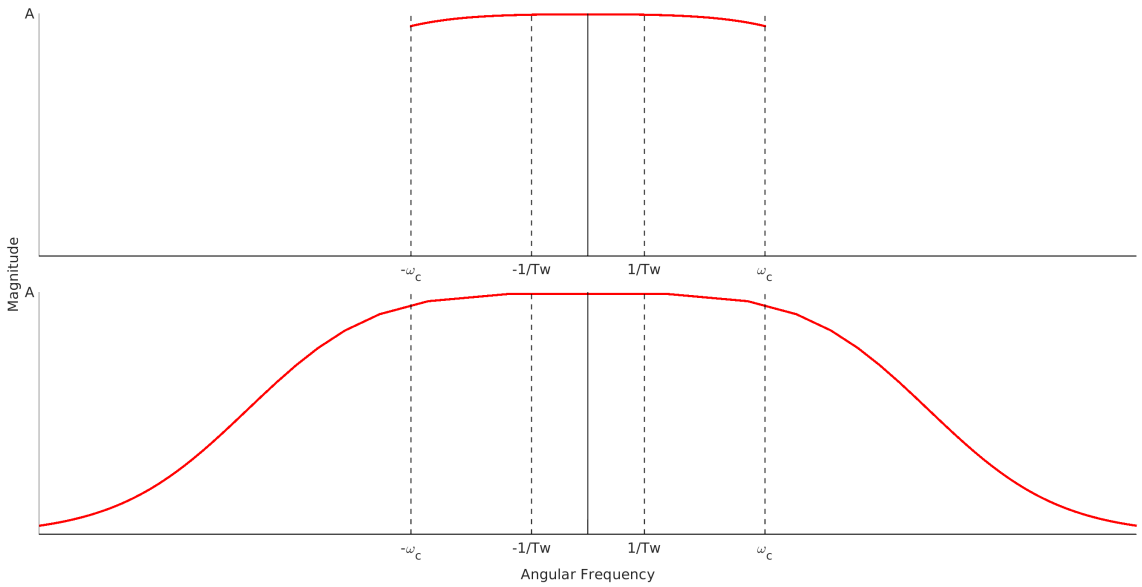

Figure 10: Drawings of the Output PSDs.

To determine $E\{y^2\}$, the integral of the PSD can be taken. Since white noise has a mean of 0 and is uncorrelated with itself, this expectation is equivalent to the variance.

$$\begin{aligned} E\{y^2 - \bar{y}^2\} &= E\{y^2\} - E\{\bar{y}^2\} = E\{y^2\} - 0 \\ &= E\{y^2\} \\ &= \int S_y(j\omega) d\omega \end{aligned}$$

For case 1:

$$\begin{aligned} E\{y^2\} &= \int_{-\omega_c}^{\omega_c} \frac{1}{T_\omega^2 \omega^2 + 1} A d\omega \\ &= \frac{A}{T_\omega} \tan^{-1}(T_\omega \omega) \Big|_{-\omega_c}^{\omega_c} \\ &= \frac{A}{T_\omega} \left(\tan^{-1}(T_\omega \omega_c) - \tan^{-1}(-T_\omega \omega_c) \right) \\ &= \frac{2A}{T_\omega} \tan^{-1}(T_\omega \omega_c) \end{aligned}$$

For case 2:

$$\begin{aligned} E\{y^2\} &= \int_{-\infty}^{\infty} \frac{1}{T_\omega^2\omega^2 + 1} A d\omega \\ &= \frac{A}{T_\omega} \tan^{-1}(T_\omega \omega)\Big|_{-\infty}^{\infty} \\ &= \frac{A}{T_\omega} \left(\tan^{-1}(T_\omega \infty) - \tan^{-1}(-T_\omega \infty)\right) \\ &= \frac{A}{T_\omega}\pi \end{aligned}$$

Through analysis of the sketches from *part b*, it can be seen that the output PSD for the pure and bandlimited white noise are very similar. As ω_c is made larger than $\frac{1}{T_\omega}$ the magnitudes at those frequencies approach zero, meaning they are less important than those frequencies inside the band of interest. When looking at the calculated variance from *part c* it can be seen that as ω_c increases σ_y^2 decreases. This again implies that if ω_c is made larger than the bandwidth of the system, it is not considerably different than the bandlimited system and little error is induced by assuming the entire spectrum is flat.