Week 1

Source Recommendations

- System Modeling and Identification by Rolf Johansson

- System Identification: An Introduction by Karel J. Keesman

- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-435-system-identification-spring-2005/

Course Outline

$$\begin{tikzcd}
 & \text{Kalman Filter} & & \\
\text{Probability} \arrow[ur] & & \text{State-space} \arrow[ul] & \\
 & & \text{Measurement} & \text{sensor} \arrow[l] \\
\text{Random Variables} \arrow[uu] \arrow[r] \arrow[dr] \arrow[dd] \arrow[ddr] & \text{scalar} & & \text{sensor fusion} \arrow[u] \\
 & \text{Vector} & & \\
\text{Mean} \arrow[d] & \text{Variance} \arrow[d] & & \\
\text{Correlation} & \text{Covariance} & &
\end{tikzcd}$$

Re-cap for Future Classes

Consider we have a RC circuit and we want to know change of charge over time. At the first stage, we should convert the problem into an algebraic equation. $V_i = V_R + V_C \;\rightarrow\; V_i = Ri + \frac{1}{C}\int i\,dt \qquad i = C\frac{dV}{dt}$

$V_i = R\dot{q} + Cq\,, q(t=0) = q_0 \qquad q(t) = (q_0 - V_iC)\,e^{-\frac{t}{RC}} + \frac{V_i}{C} \qquad \dot{q} = -\frac{1}{RC}q + \frac{V}{R}$

At this point we have a differential equation but, we can alter it to an algebraic with a transfer function. To find the transfer function take the Laplace transform of the differential equation. We can use analogies to simplify this process.

$$\boxed{\underbrace{\dot{x} + ax}_{\text{Homogenous}} = \underset{\text{Forcing term}}{b}}$$

$$sX(s) + aX(s) = B(s) \;\rightarrow X(s) = \frac{1}{s+a}B(s)$$

Adapt the change of charge equation to this equation,

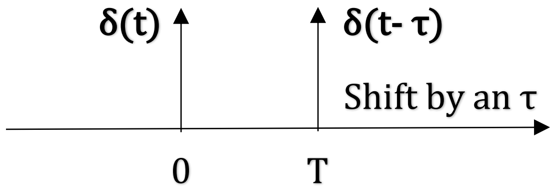
$$Q(s) = \boxed{\frac{1}{s+\frac{1}{RC}}} V_i(s)$$

Transfer function

First order linear system

Note: In transfer functions, there are two terms which are zeros and poles. Zeros are roots of numerator and poles are roots of denominator.

If R and C values are not known, Delta-Dirac or Impulse function is helpful since it's Laplace transform is 1.



$\frac{dH}{dt} = \delta(t) \int_{-\infty}^{\infty} \delta(t) = [H(t)]^{\infty} = 1 - 0 = 1$ $L\{\delta(t)\} = \int_{-\infty}^{\infty} e^{-st}\delta(t)dt = \int_{t\neq 0} e^{-st}.0dt + [e^{-st}.\delta(t)]\,(t=0)$
$= 0 + [e^{0}.\delta(0)]$ $L\{\delta(t)\} = 1$ $\boxed{Q(s) = G(s)\,V_i(s)}$

Output Input

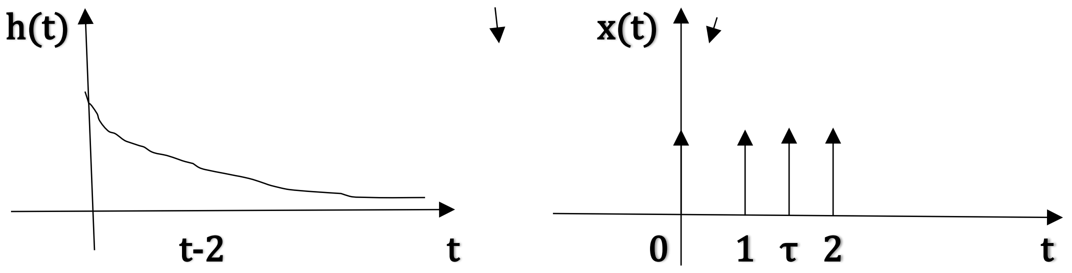
When the input is impulse function, estimation of R and C will be no longer a trouble.

Method of identification is encumbered by:

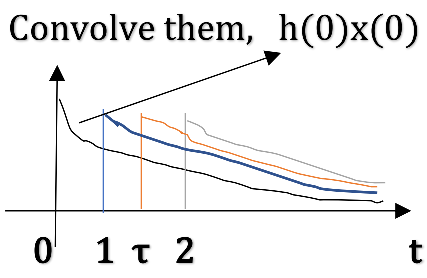
- Noise. There are two sources of noise. One of the is measurement and other one is process (like components in circuit).

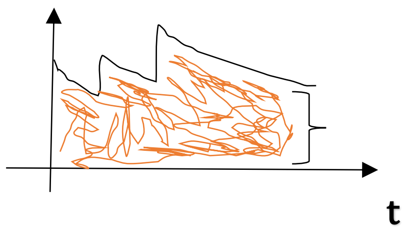
- δ(t) is not a physical reality but useful.

Convolution is a method for creating an output signal from two input signals. For example, convolve h(t) with a Delta Comb



Convolve them,

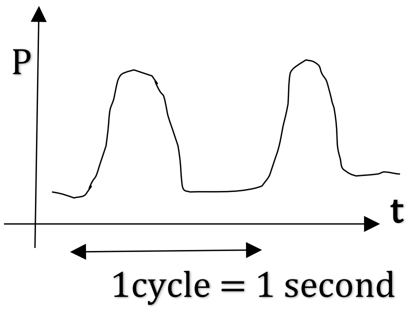




To compute area under the curve, using integral is reasonable.

$$\boxed{y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau} \longrightarrow \text{dummy variable}$$

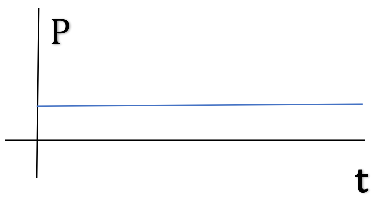
Continuous time is what we live in but, we use computers to model complex system thus, we should learn how they see the time. Let's understand with an example.



Heart rate = 60 beats per minute ideally.

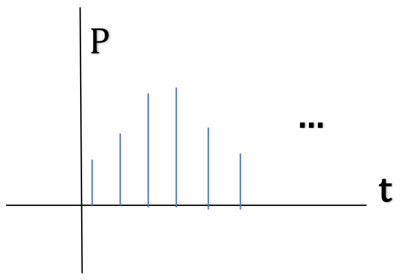
It is in continuous time but, how computers see ? We can convert this with different period time. (Analog to Digital)

If T=1 second,



This graph does not give a plausible knowledge

Getting period lower and lower gives us an opportunity to interpret.



Now, the graph is more understandable.

Homework

$$ae^{-bt} * Asin(\omega t)$$

Convolve these two signals in three ways.

- Convolution integral & Laplace transform & MATLAB

Homework Answer

$-\ 1^{st} way$

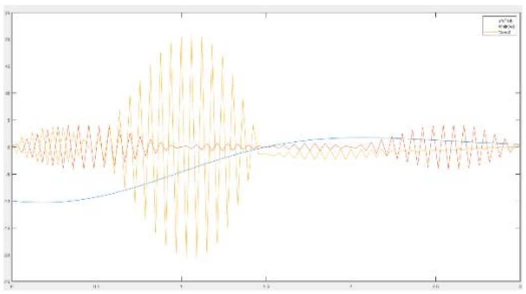
$ae^{-bt} * Asin(\omega t) = \int_{-\infty}^{\infty} ae^{-b\tau}\, Asin(\omega(t-\tau))\, d\tau = \int_{-\infty}^{\infty} e^{-b\tau}\, [sin(\omega t)cos(\omega\tau) - cos(\omega t)sin(\omega\tau)]\, d\tau$

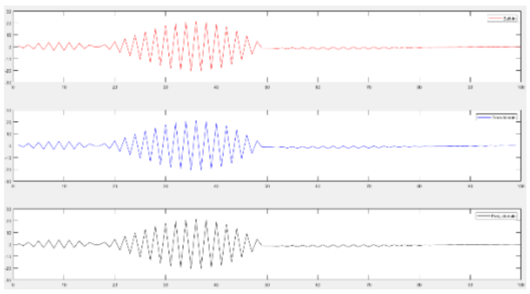
$= 2sin(\omega t)\int_{0}^{\infty} cos(\omega\tau)e^{-b\tau}\, d\tau \quad = \frac{-2ab}{b^2+\omega^2} Asin(\omega t)$

$-\ 2^{nd} way$

$L\{ae^{-bt}\} = \frac{a}{s+b}$ $\quad$ $L\{Asin(\omega t)\} = A\frac{\omega}{s^2+\omega^2}$ $\quad$ $L\{ae^{-bt} * Asin(\omega t)\} = \frac{a}{s+b}\ A\frac{\omega}{s^2+\omega^2}$

$-\ 3^{rd} way$





Figures. F and G signals and their convolution & Comparison of Conv.

Week 2

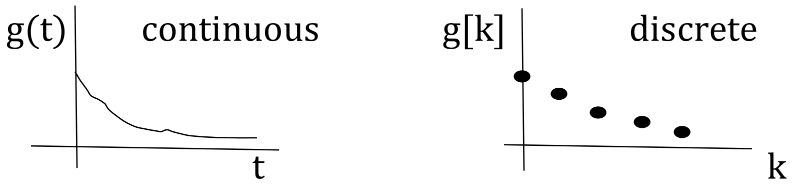
Signal Systems

$$\underbrace{B(s)}_{\text{Input}} \longrightarrow \underbrace{\boxed{G(s)}}_{\text{System}} \longrightarrow \underbrace{Y(s)}_{\text{Output}} \qquad \boldsymbol{Y(s)=\frac{1}{s+A}[y_0+B(s)]}$$

y_0 is the initial condition and we determined as it is Delta-Dirac function.

$B(s)\ is\ the\ step\ function\ so, \boldsymbol{B(s)=\frac{B}{s}}$ $\qquad$ $\boldsymbol{Y(s)=\frac{y_0}{s+A}+\frac{B}{s(s+A)}}$

Important Note: Impulse response is the system itself.



We have seen convolution in continuous time, what about convolution in discrete time ?

Example:

$$\boldsymbol{u[k]=[0\ 1\ \ 0]\ \ h[k]=[4-2\ \ 3]}$$

$$\boldsymbol{y[k]=u[k]*h[k]=?}$$

$$\boldsymbol{y(t)=\int_{\tau=0}^{\tau=\infty} h[\tau]\,u[t-\tau]d\tau=\int_{\tau=0}^{\tau=\infty}(t-\tau)\,u[\tau]d\tau}$$

$$\boldsymbol{y[kT]=y[k]=\sum_{m=0}^{m=\infty} h_m u_{k-m}=y_k}$$

K	
0	$\boldsymbol{y[0]=\sum_{m=0}^{m=\infty} h_m u_{-m}=0}$
1	$\boldsymbol{y[1]=\sum_{m=0}^{m=\infty} h_m u_{1-m}=4}$
2	$\boldsymbol{y[2]=\sum_{m=0}^{m=\infty} h_m u_{2-m}=-2}$
3	$\boldsymbol{y[3]=\sum_{m=0}^{m=\infty} h_m u_{3-m}=3}$

Random Variables

Random variable takes random number is called a processing. All the processing are called ensemble. Stochastic is also called Randomness and it is opposite of Deterministic. We have to deal with stochastic variables since, the noise is stochastic.

If there is no significant difference in output or realization while time shifts, it is called strictly sense stationary(SSS). And weak sense stationary (WSS) is a condition which the covariance of the data does not change respect to time. For example, white noise is WSS.

Probability Density Function (PDF) describes all the possible realizations of Random Variables(RV).

Important Note: Probability and possibility are different terms. Probability is about the area. $\boldsymbol{F(x) = \int_a^b pdf\ dx}$

Expectation can be described as a mean but it is slightly different. Mean is extracted after the experiment while, expectation is calculated before the experiment. Expectation is applied to stochastic process as, in deterministic process, we do not expect anything, we just know it.

$\boldsymbol{E[X] = \sum_{i=1}^{k} x_i p_i = x_1p_1 + x_2p_2 + \cdots + x_kp_k}$ (x is outcome, p is probability.)

Example – Rolling a Dice - :

Probability of each condition is $\boldsymbol{\frac{1}{6}}$

$$\boldsymbol{E(X) = 1 * P(1) + 2 * P(2) + \cdots + 6 * P(6)}$$

$$\boldsymbol{E(X) = \frac{7}{2} = 3.5}$$

Homework

Write a MATLAB code to plot Gaussian PDF and calculate covariance.

Homework Answer

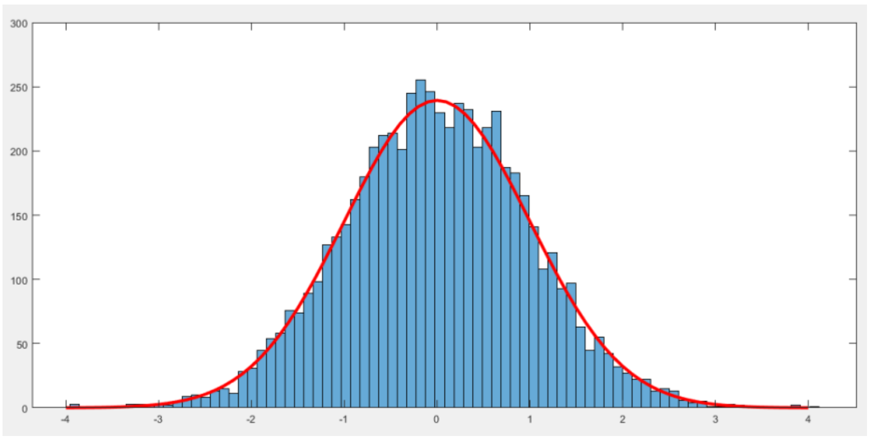


Figure. PDF of Random Variable

Week 3

Correlation & Covariance

Correlation is a relationship between two random variables. Expected value is introduced in previous lecture and we can find correlation coefficient by expectation.

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E(X)^2} \, . \sqrt{E(Y^2) - E(Y)^2}}$$

Also covariance can be used to determine correlation coefficient. But what is covariance ? Covariance is a measure of the variability of two random variables. $cov(X,Y) = E[(X - E[X])(Y - E[Y])]$

Important Notes: Delta comb determines intervals in discretization method.

White noise' s mean is zero. So it's expectation is also zero. Correlation is covariance for white noise.

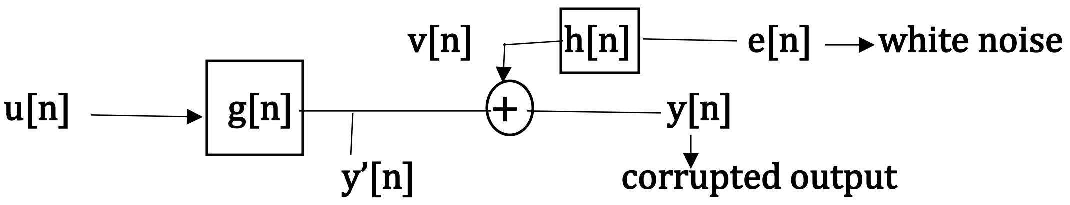
Dealing with Noise

If we do not want to deal with noise, convolve the signal and system;

$u[n] * h[n] = \sum_{l=0}^{\infty} h[l]u[n-l]$ **(l is dummy variable , n is real time)**

$R_{uh} = \sum_{m=0}^{\infty} u^T[n]h[n-m]$ **(m is dummy variable , n is real time)**

Convolution appears once while correlation twice in real time.



Signal, system, matrix, vector and filter are interchangeable terms. Let's say, **g[n]** is system filter and **h[n]** is noise filter.

$$y[n] = \underbrace{\sum g[m]u[n-m]}_{y'[n]} + \underbrace{\sum h[l]u[n-l]}_{v[n]}$$

This representation was the goal of what we have learnt so far.

To filter noise from the signal autocorrelation method can be used too.

$$R_{XX}(t_1, t_2) = E[X_{t_1}\overline{X}_{t_2}]$$

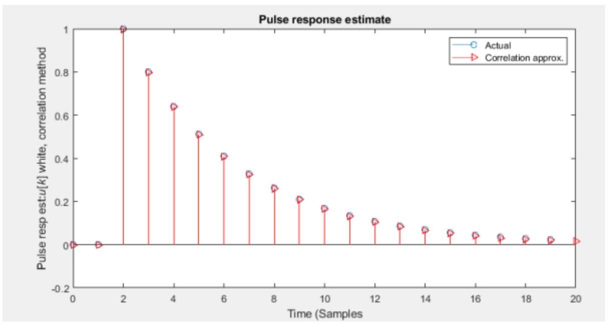
And autocovariance is obtained by subtracting the mean.
$K_{XX}(t_1, t_2) = E[(X_{t_1} - \mu_{t_1})\overline{(X_{t_2} - \mu_{t_2})}$

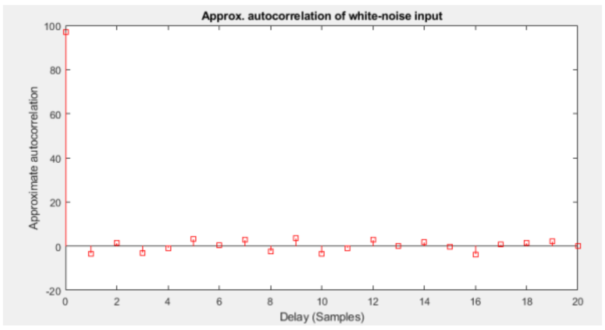
Homework:

$$\boldsymbol{R_{yu}^{N}[\tau] \cong \sum_{m=1}^{N} g_0[n]\, R_u^{N}[\tau - n]}$$

Write a MATLAB script and display estimation of impulse response and autocorrelation of the white noise regarding this equation.

Homework Answer:





Figures. Plot of Pulse Response Estimate & Plot of Autocorrelation of White Noise Input

Week 4

Time Shift

Time shift is an operator like derivation. It means this operator supplies estimation of the next signal due to all past signals.

$q\{u[k]\} = u[k+1] \qquad q^{-1}\{u[k]\} = u[k-1]$

Apply this principle to Impulse Response, $y[k] = \sum_{m=1}^{\infty} g_0[m]\, u[k-m]$

Now impulse response becomes system filter, $G(q) = \sum_{m=1}^{\infty} g_0[m]\, q^{-m}$

Reminder: $H(R) = \sum^{\infty} p_0^n\, R^n$ (p_0 is impulse response or system) $R \rightarrow \frac{1}{z}$

$$H(z) = \sum g[n] z^{-n}$$

Apply same method to noise, $y[k] = \sum_{m=1}^{\infty} q_0[m]\, u[k-m] + v[k] = G(q)u[k] + H(q)e[k]$

Sinusoidal Input without Noise

	Continuous Time	Discrete Time
Periodic	Fourier Series (FS)	DFT
Non - Periodic	Fourier Transform (FT) $\rightarrow$	DTFT

Delta comb version

Important Note: Fourier Transform is used with $T = \infty$ to make periodic, by this, it can not be disapproved.

Reminder: Euler's Identity $\rightarrow e^{j\omega k} = cos\omega k + jsin\omega \quad \rightarrow u[k] = cos\omega k = Re[e^{j\omega k}]$

Apply this principle to Impulse Response (system output),
$y[k] = \sum_{m=1}^{\infty} g[m]\, u[k-m]$

$g[m]$ is a real system. $y[k] = Re\{e^{j\omega k} \sum_{m=1}^{\infty} g[m]\, e^{-j\omega m}\}$

Define frequency function, $G(e^{j\omega}) = \sum_{m=1}^{\infty} g[m]\, e^{-j\omega m}$ $\quad y[k] = G(e^{j\omega}) Re\{e^{j\omega k}\} = G(e^{j\omega})\, cos(\omega k + \varphi)$

Important Note: $G(e^{j\omega})$ *is eigenvalue* $\qquad cos(\omega k + \varphi)$ *is eigenvector*

Phasor is time-independent complex vector.

$G(j\omega)\begin{cases} < 1 & attenuates \\ > 1 & amplify \end{cases} \qquad \varphi \begin{cases} < 0 & lagging \\ > 0 & leading \end{cases}$

Sinusoidal Input with Noise

$$y[k] = \alpha G(e^{j\omega}) \cos(\omega k + \varphi) + v[k]$$

Purpose is get rid of v[k] – noise – . Assume that v[k] is white-noise.

To find magnitude and phase of the signal,

$$I_C[N] = \frac{1}{N}\sum_{k=1}^{N} y[k]\cos(\omega k) \qquad I_S[N] = \frac{1}{N}\sum_{k=1}^{N} y[k]\sin(\omega k)$$

Substitute to $I_C[N]$,

$$I_C[N] = \frac{1}{N}\sum_{k=1}^{N} \alpha\, G(e^{j\omega})\left(\frac{1}{2}\big(cos(\omega k + \varphi + \omega k) + cos(\omega k + \varphi - \omega k)\big)\right) + \boxed{\frac{1}{N}\sum_{k=1}^{N} v[k]\cos(\omega k)}$$

It is Expectation of white-noise. Since expectation of white-noise is zero, the noise of the system becomes zero.

After neglection of small values, $I_C[N] = \frac{\alpha}{2}\, G(e^{j\omega})\cos(\varphi)$

With similar process, $I_S[N] = \frac{\alpha}{2}\, G(e^{j\omega})\sin(\varphi)$ $\qquad \varphi[\widehat{G}(e^{j\omega})] = tan^{-1}\frac{I_S[N]}{I_C[N]}$

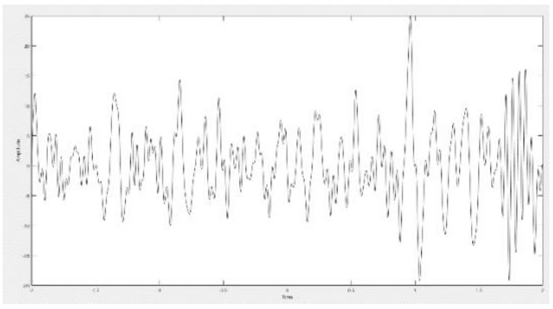
Auto-covariance

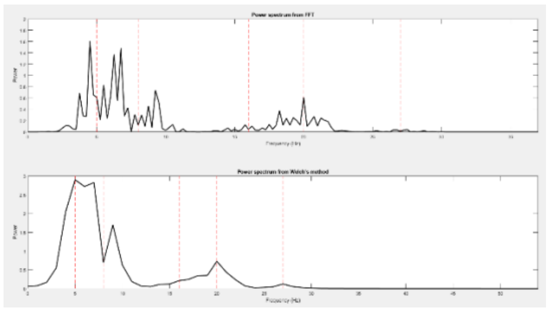
$$cov[X_{t_1}, X_{t_2}] = E[X_{t_1}X_{t_2}] - \mu_{t_1}\mu_{t_2}$$

Homework:

Plot Power Spectrum from Fourier Transform and Welch's method on MATLAB.

Homework Answer:





Figures. Plot of the Created Signal & Power Spectrum with two methods

Week 6

Cross-Correlation

$R_{uy} \neq R_{yu}$ $\quad R_{uu}[m,n] = E\{u[m]\, u[n]\}$ If we put t to m, it is going to be correlation, if we put t to n, it is going to be covariance. If u is wide-sense stationary, time is not important in random process.

$R_{yu}[\tau] = \sum g[m]\, u[k-m]\, u[k-\tau]$ $\qquad R_u[\tau] = E\{u[k]u[k-\tau]\}$

τ comes from, $k-(k-\tau)=\tau$

$R_{yu}[\tau] = \sum g[m]\, R_u[\tau-m]$ $\quad R_{yu}[\tau] = g[\tau] * R_u[\tau]$ $\quad R_{uy}[\tau] = \sum g[m]\, R_u[\tau+m]$

$R_{uy}[\tau] = g[-\tau] * R_u[\tau]$ $\quad R_y[\tau] = g[\tau] * R_{uy}[\tau]$ $\quad R_y[\tau] = g[\tau] * g[-\tau] * R_{uy}[\tau]$

DFT of this, $\quad \varphi_y(\omega) = G(e^{j\omega})\, G(e^{-j\omega})\, \varphi_u(\omega)$

Note on Periodogram:

Periodogram is the squared magnitude of the FT of a signal it means, it is a prediction of the spectral density of a signal. $S\left(\frac{k}{NT}\right) = \left|\sum_n x_N[n]\, e^{-j2\pi\frac{kn}{N}}\right|^2$

Note on Causal and Non-causal progression:

In causal system, outputs are derived from past signals, time goes through positive direction as in real-life. But non-causal system works vice versa and it is helpful for applications like image processing or sound recording.

Bartlett -Welch's Methods

There are two well-known approaches for spectral density estimation. Bartlett method averages periodograms but these periodograms reduce variance and resolution. In Welch's method, signals are converted to frequency domain and this gives reduction of noise and frequency resolution. Since noise is reduced, Welch's method is more preferred.

State-Space Reminder:

$m\ddot{x} + c\dot{x} + kx = F_{ext}$ $\quad z_1 = x_1, z_2 = \dot{z}_1, z_3 = \dot{z}_2 = \ddot{z}_1$ $\quad m\dot{z}_2 + cz_2 + kz_1 = F_{ext}$

$\dot{z}_2 = -\frac{k}{m}z_1 - \frac{c}{m}z_2 + \frac{1}{m}F_{ext}$ $\qquad \frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\frac{F_{ext}}{m}$

Transfer Function Identification

We can describe a system respect to transfer functions. Let's look at how?

$$\begin{tikzcd} y[k] \arrow[d] & = & G(q) \arrow[drrr] & u[k] \arrow[d] & + & H(q) \arrow[d] & e[k] \arrow[dr] & \\ \text{Output} & & & \text{input} & & \text{transfer fnc} & & \text{random variables} \end{tikzcd}$$

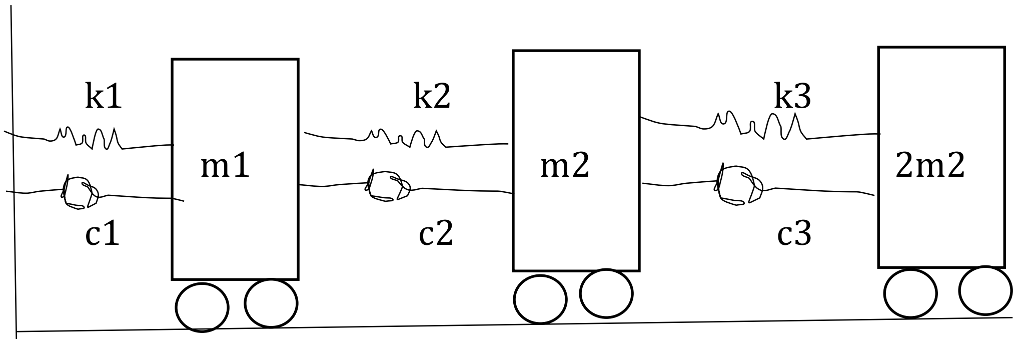
It can be written as a difference equation, $y[k] = \alpha y[k-1] + \beta u[k-1]$

In operator form, $(1 - \alpha q^{-1}) y[k] = \beta q^{-1} u[k]$ $y[k] = \frac{\beta}{q-\alpha} u[k]$

Since $(q - \alpha)$ is G(q) transfer function, we have defined the system respect to transfer function.

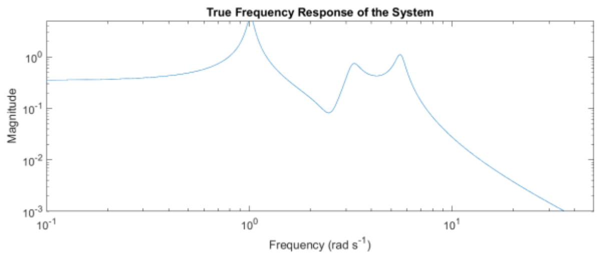
Important Note: System response is defined by pole qualitatively and by zeros quantitively.

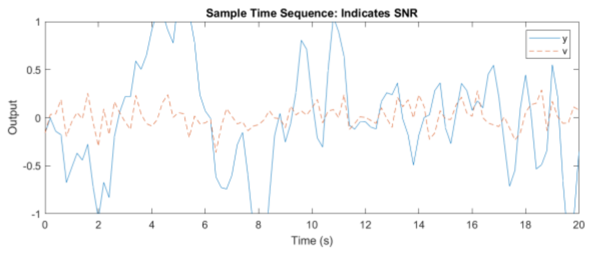
Homework:



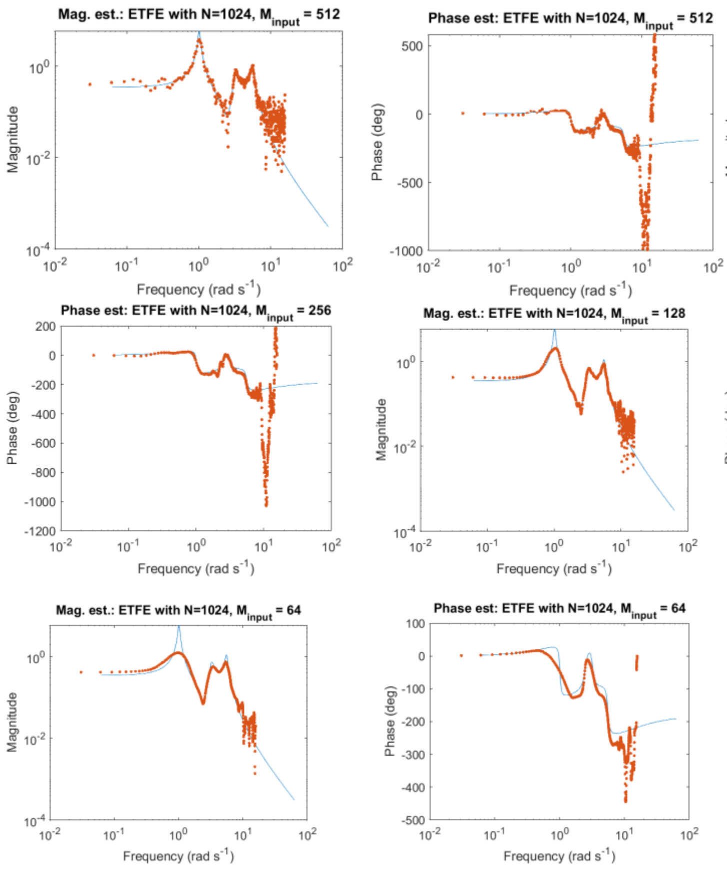
Derive system response in MATLAB. (External Force = 1)

Homework Answer:

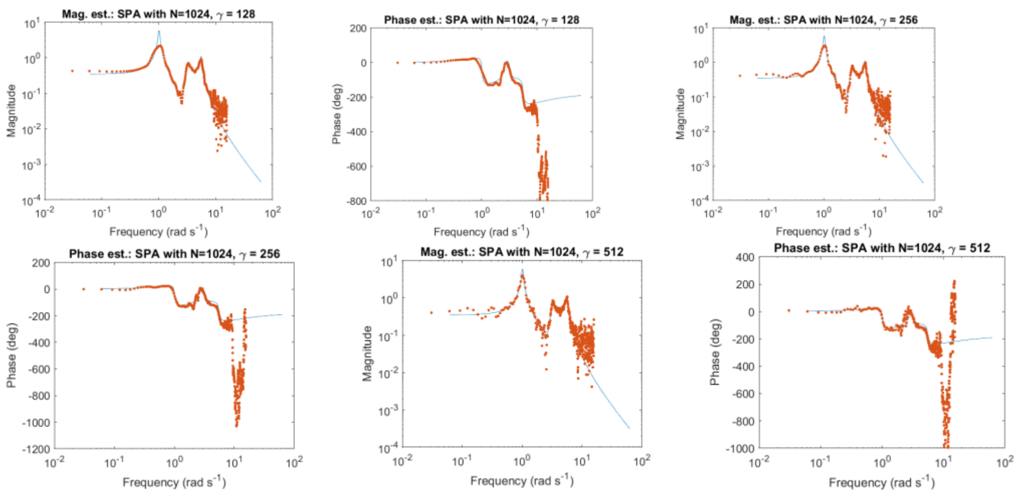




Figures. Plot of Frequency Response & Sample Time Sequence



Figures. Plots of ETFE



Figures. Plots of SPA

Week 7

Prediction Process

Moving Average: Next output is a weighted average of past inputs and there is no feedback input in the model.
Auto Regression: Past output is subtracted from last input for estimating next output. And there is a feedback output to calculate output itself.

ARMA: It is the combined method of auto regression and moving average process therefore, it is most accurate process to estimate among them.

Example:

$v[k+1] + 0.8v[k] = \frac{1}{\sqrt{\lambda}}e[k+1] + \frac{1}{\sqrt{\lambda}}0.5e[k]$ $\qquad$ $v_{k+1} = \frac{1}{\sqrt{\lambda}}[e_{k+1} + 0.5e_k] - 0.8v[k]$

To increase accuracy of prediction, decreasing e_{k+1} is essential and it is done by adjusting constants – a and c values -. (Least Square Error)

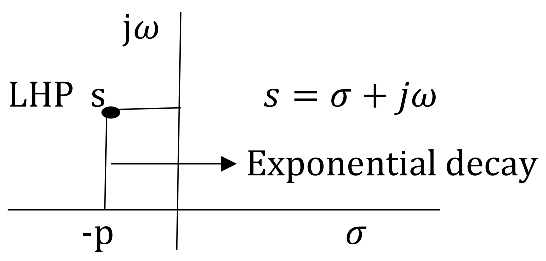
Transfer Function Identification

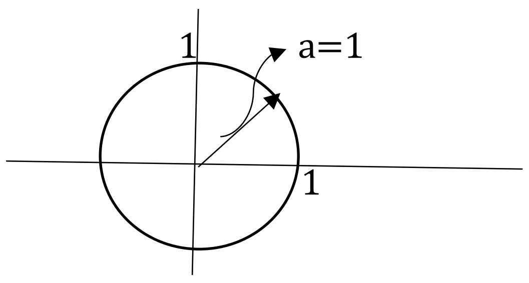
$$\tau\ (continous) \leftrightarrow a(discrete)$$

$$\text{time constant} \left\{ \begin{array}{l} small \rightarrow fast \\ big \rightarrow slow \end{array} \right\} \begin{array}{l} \text{system} \\ time\ constant\ \equiv base\ of\ series \end{array}$$

In damped sinusoid, if a = 1 there will remains only cos ωk so, the system becomes eternal sinusoid.

Time (s-domain) $\qquad\qquad$ Discrete (q-domain)





The higher in j axis you are, the faster oscillations you get. In continuous time, solution of second order system is like that; $x(t) = Ae^{-\zeta\omega_n t}\cos(\omega_d t + \theta)$
ζ *is damping ratio*, ω_n *is natural freq.*, ω_d *is damped freq.*

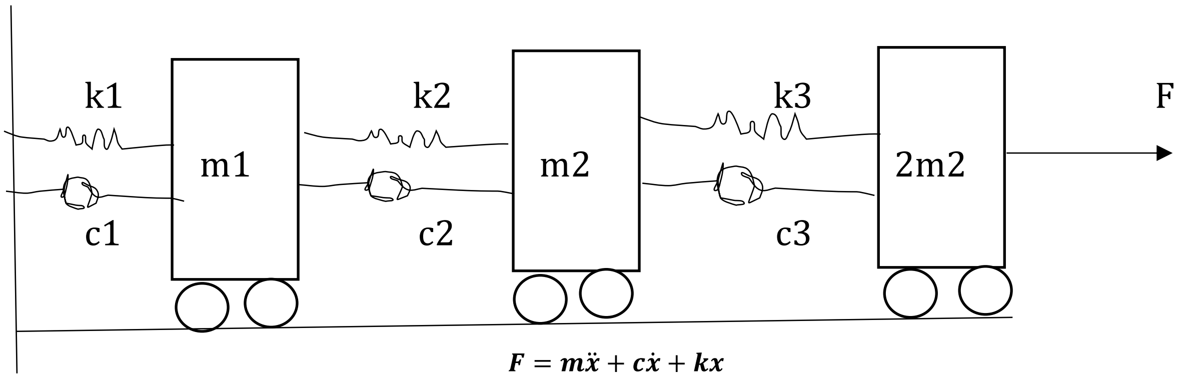
In discrete time, solution becomes that; $g(t) = u(t)e^{-\sigma t}\cos(\omega_c t)$

To make a connection between discrete and continuous time,

$t \rightarrow kT_s, u(t) = 1[k] \quad a = e^{-\sigma T_s} \quad \omega_d = \omega_c T_s \quad q = e^{sT_s}$

Week 8

MID-TERM EXAM



$$\boldsymbol{F = m\ddot{x} + c\dot{x} + kx}$$

Since the equation is second order differential, we should have two state variables; $\boldsymbol{x_1 = x}$ $\boldsymbol{x_2 = \dot{x}}$ $\boldsymbol{\dot{x_1} = \dot{x} = x_2}$ $\boldsymbol{\dot{x_2} = \ddot{x}}$

Now, we can adjust our equation to first order,

$\boldsymbol{m\dot{x_2} + cx_2 + kx_1 = F}$ $\boldsymbol{\dot{x_2} = -\frac{c}{m}x_2 - \frac{k}{m}x_1 + \frac{F}{m}}$

Write two state equation in a matrix format,

$$\boldsymbol{\begin{Bmatrix}\dot{x_1}\\ \dot{x_2}\end{Bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{k}{m} & -\frac{c}{m}\end{bmatrix} . \begin{Bmatrix}x_1\\ x_2\end{Bmatrix} + \begin{bmatrix}0\\ \frac{1}{m}\end{bmatrix} . \{F\}}$$

State matrix Input matrix

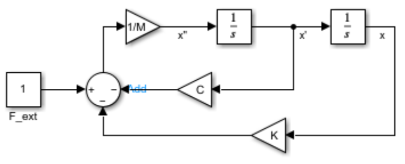
To get output and direct transmission matrices,

$$\boldsymbol{y = x_1} \qquad \boldsymbol{\{y\} = [1\ 0] . \begin{Bmatrix}x_1\\ x_2\end{Bmatrix} + [0] . \{F\}}$$

Output matrix Direct transmission matrix

We have multiple spring, damper and mass values, let's put them in matrices,

$$M = \begin{bmatrix}m_1 & 0 & 0\\ 0 & m_2 & 0\\ 0 & 0 & 2m_2\end{bmatrix} \quad K = \begin{bmatrix}k_1 + k_2 & -k_2 & 0\\ -k_2 & k_2 + k_3 & -k_3\\ 0 & -k_3 & k_3\end{bmatrix} \quad C = \begin{bmatrix}c_1 + c_2 & -c_2 & 0\\ -c_2 & c_2 + c_3 & -c_3\\ 0 & -c_3 & c_3\end{bmatrix}$$



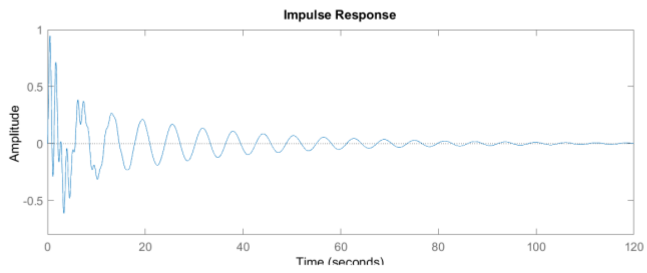


Figure 1-2. Block Diagram of the System & Impulse Response of the System

Comment: As expected impulse response of the system is damping over time. Transfer function of 2nd order systems; $H(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$

Substitute damping ratio and natural frequency for our system,

$\omega_n = \sqrt{\frac{K}{M}}$ $\zeta = \frac{\omega_n C}{2K}$

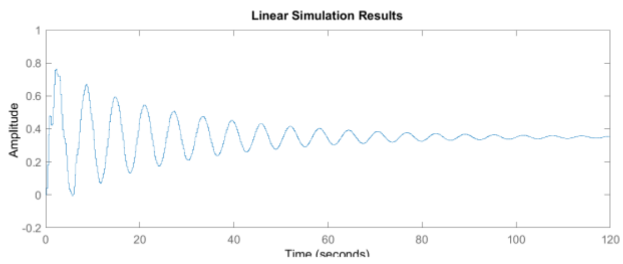


Figure 3. Discrete-time Simulation

Comment: System is discretized. Difference between impulse response and discrete response comes from impulse response method.

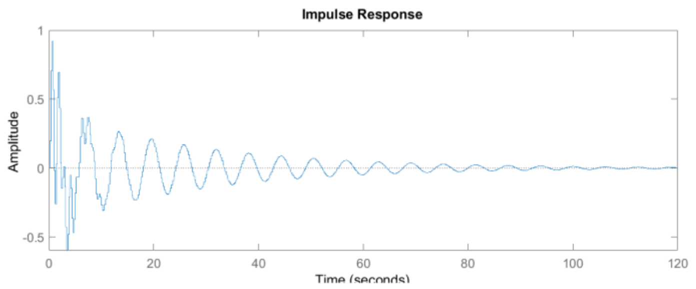


Figure 4. Impulse Response of Discrete-time System

Comment: Impulse response in discrete and continuos time is equal as expected.

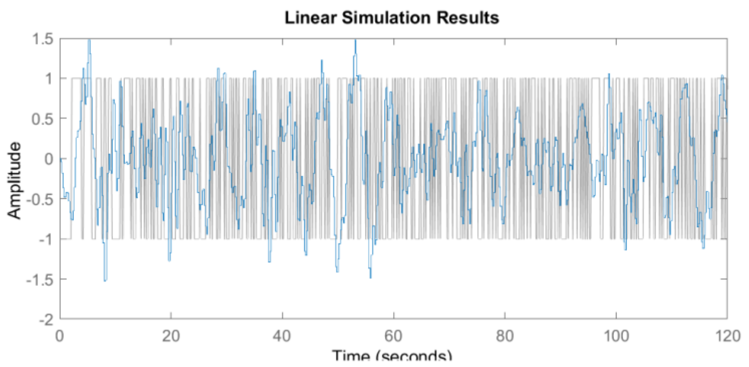


Figure 5. Discrete-time Simulation with White-noise

Comment: Signal is discretized and white-noise is added.

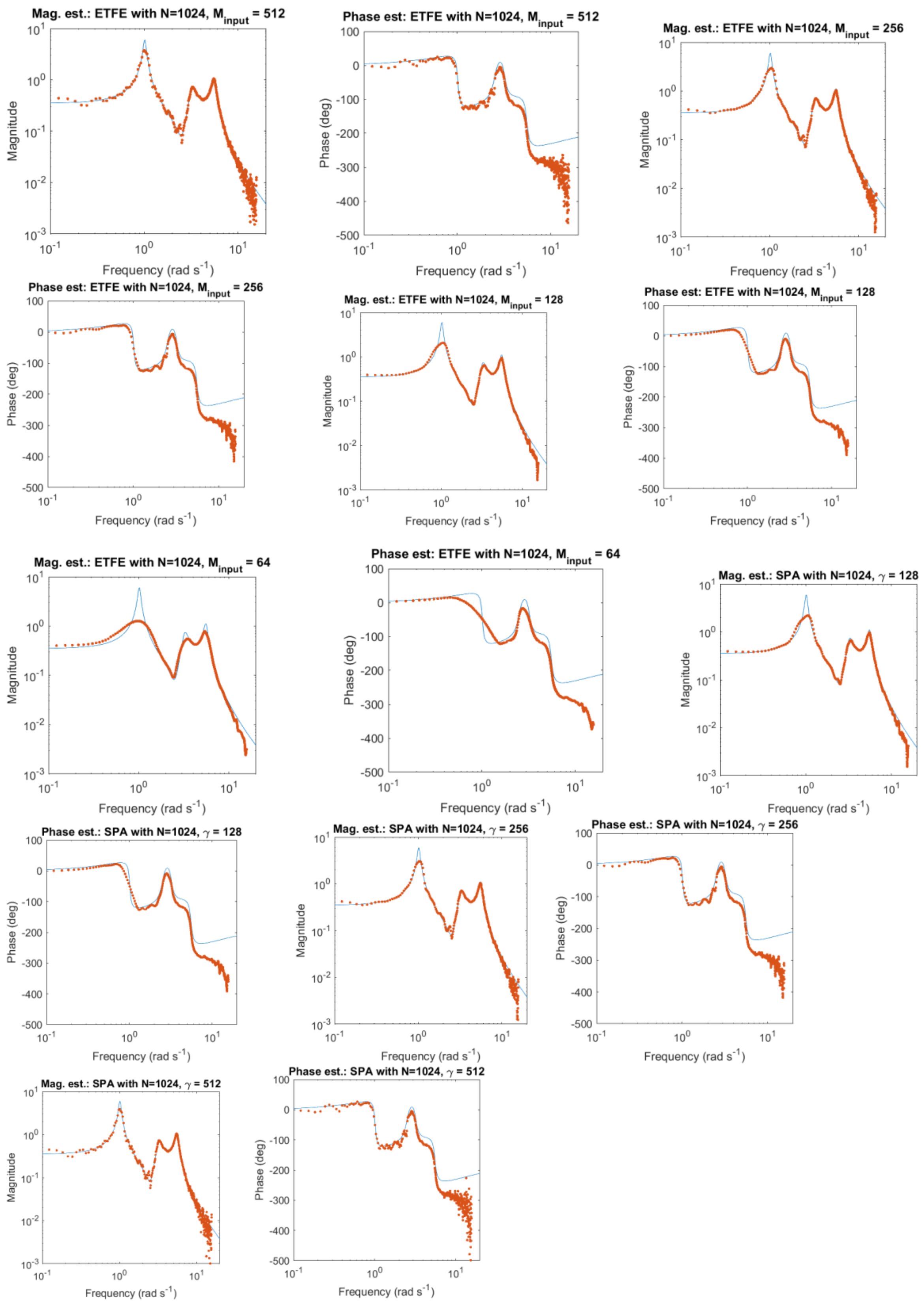


Figure 6. System Estimations ETFE and SPA for different inputs

Comment: In ETFE, while M_{input} decreases, scatters becomes smother but dynamics of the system blurring. The relation is same with γ in SPA.

Week 9

q-Domain

As you increase in imaginary axis (get more data), you get more oscilations

When q=1 there is no damping

Correspondence with continuous- time signals

$s = \sigma + j\omega_{continuos}$ $\sigma = -\zeta\omega_{natural}$ $\omega_d \equiv \omega_{continuos} = \omega_{discrete} = \omega_n\sqrt{1-\zeta^2}$

In q -Domain, $\zeta = -\frac{\ln[r]}{\sqrt{\theta^2 + \ln^2 r}}$

σ is in left hand plane for stability. And while r is decreases, θ is increasing.

In top expressions ζ is constant, what about θ is constant ? $\omega_d = \frac{1}{T_s}\theta$

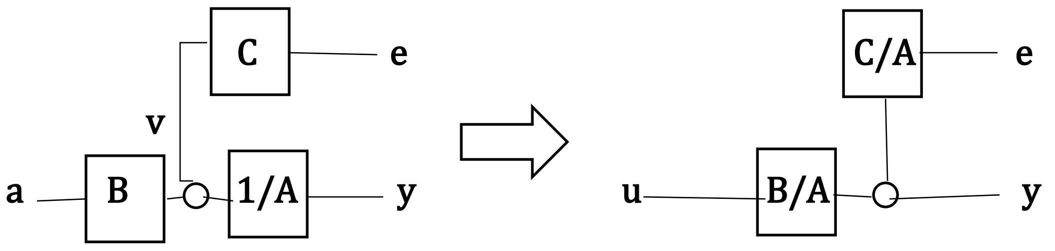
w-Transform

It is like s to q. q to w. But it is tangential. q-Domain is important for us because, we take sample in continuous but work in discrete.

Standard Model Forms

Poles must be less than 1 for stability in q-Domain.

ARMAX Model



$B = b_1q^{-1} + \dots + b_{nb}q^{-nb}$ $A = 1 + a_1q^{-1} + \dots + a_{na}q^{-na}$ $C = 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc}$

Difference eq: $y[k] + a_1y[k-1] + \dots + a_{na}y[k-n_a] = b_1u[k-1] + \dots + b_{nb}u[k-n_b] + e[k] + c_1e[k-1] + \dots + c_{nc}u[k-n_c]$

ARX Model

Simplest and most common model for system estimation. $\mathrm{Ay = Bu + e}$

Difference eq: $y[k] + a_1y[k-1] + \dots + a_{na}y[k-n_a] = b_1u[k-1] + \dots + b_{nb}u[k-n_b] + e[k]$

Simulation or Prediction

$$y[k] = G(q)u[k] + H(q)e[k], \textit{where } e[k] \textit{is white noise}$$

$$y[k] = \{1 - H^{-1}(q)\}y[k] + H^{-1}(q)G(q)u[k] + 0$$

So we do not need future outputs to determine present we just need past inputs and outputs.

Define cost function

$$V_n(\boldsymbol{\theta}) = \sum_{k=1}^{n} y[k]^2 - \underbrace{2\boldsymbol{\theta}^T \sum_{k=1}^{n} x[k]y[k]}_{R_{xy}} + \underbrace{\boldsymbol{\theta}^T \left\{\sum_{k=1}^{n} x[k]x^T[k]\right\} \boldsymbol{\theta}}_{R_{xx}}$$

$$y = \theta_1 x_1 + \theta_2 x_2 + \varepsilon$$

Homework:

Write a MATLAB script for Dryer Example.

Homework Answer:

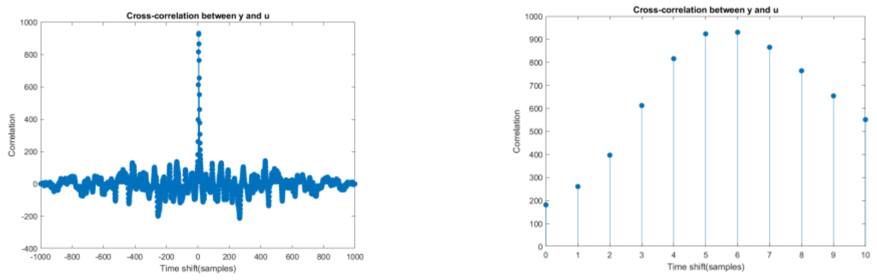
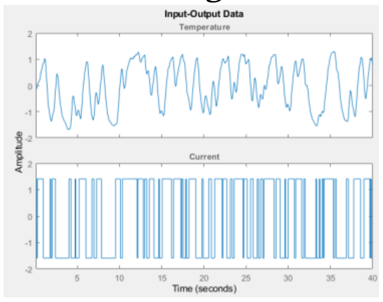


Figure 1. Cross Correlation of input and output



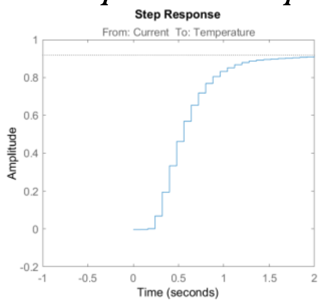
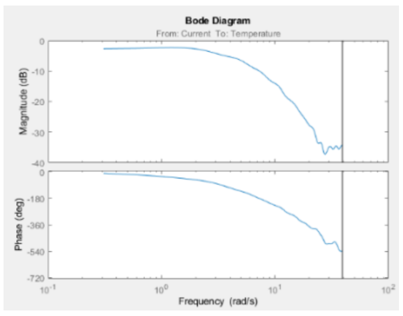


Figure 2-3. Input-Output Data & Step Response



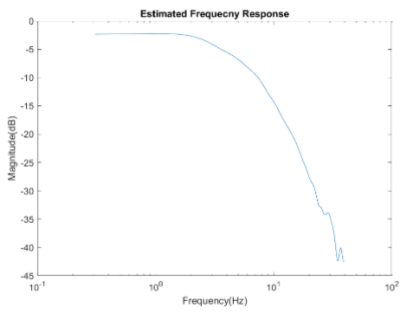
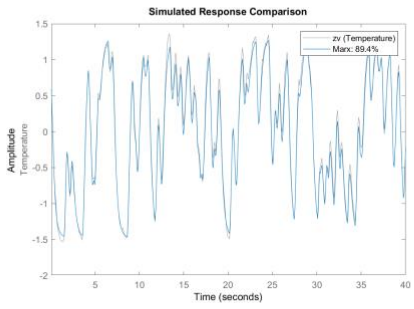


Figure 4-5. Bode Diagram & Estimated Frequency Response



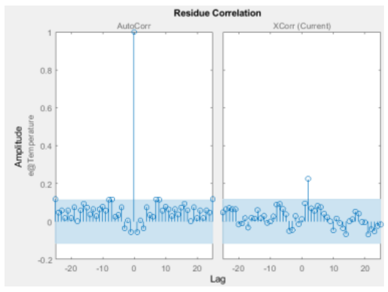
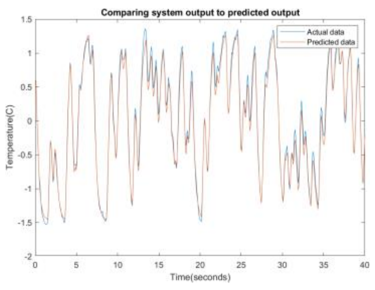


Figure 6-7. Simulated Response Comparison & Residue Correlation



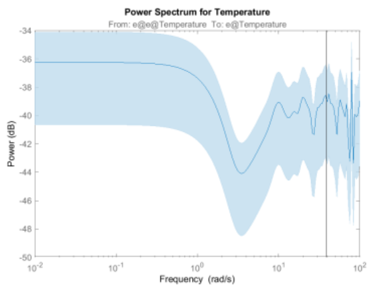


Figure 8-9. Comparing system output to predicted output & Power Spectrum for Temperature

Week 10

Principal Component Analysis

When we have data have multiple channels, to see the correlation we need PCA. Since we can only display 3-D at most, we must reduce dimensions and that's what PCA does. In PCA, where the data variance is highest, the importance is the most and vice versa.

A good explanation: https://www.youtube.com/watch?v=FgakZw6K1QQ

Bayes Theorem

Start with define false positive and false negative with an Covid-19 example.

False positive: test is positive but you are not sick. False negative: test is negative but you are sick. And we have two terms which are essentials to determine top. Prevalence: probability of being sick. Accuracy: probability of positive test result with being sick. Go to calculations;

$$P(positive) = P(accurate) * P(sick) + P(inaccurate) * P(healthy)$$

$$P(negative) = P(accurate) * P(healthy) + P(inaccurate) * P(sick)$$

P(sick | positive), the probability to be sick if you test positive

P(healthy | negative), the probability to be healthy if you test negative

Now we need Bayes rule: P(A | B) = P(B | A) * P(A) / P(B)

$$P(positive) = P(accurate) * P(sick) + P(inaccurate) * P(healthy)$$

P(sick | positive) = P(positive | sick) * P(sick) / P(positive)

= accuracy * prevalence / P(positive)

P(healthy | negative) = P(negative | healthy) * P(healthy) / P(negative)

= accuracy * (1-prevalence) / P(negative)

Intro. To Kalman Filter

We have mentioned signal, system and noise earlier. Noise is what we don't want as engineers and filters are used to reduce these noise. One of the well-known filter for linear systems is Kalman filter and it also does estimation. State equation is ; $\boldsymbol{x_{k+1}} = \boldsymbol{Ax_k} + \boldsymbol{Bu_k} + \boldsymbol{w_k}$

Where x is state variable, u is input and w is process noise

Output equation is; $\boldsymbol{y_k} = \boldsymbol{Cx_k} + \boldsymbol{z_k}$ where y is output, z is measurement noise.

To apply Kalman filter, we have to make some assumptions;

Average value of w and z is zero & No correlation between w and z

Then define covariance matrices, $\boldsymbol{S_w = E(w_k w_k^T)} \quad \boldsymbol{S_z = E(z_k z_k^T)}$

Now filtering, $\boldsymbol{K_k = AP_kC^T(CP_kC^T + S_z)^{-1}}$

Where K is Kalman gain and P is estimation error covariance.

$$\boldsymbol{\hat{x}_{k+1}} = \underbrace{(\boldsymbol{A\hat{x}_k + Bu_k})}_{\text{state estimate}} + \underbrace{\boldsymbol{K_k(y_{k+1} - C\hat{x}_k)}}_{\text{correction term}}$$

$$\boldsymbol{P_{k+1} = AP_kA^T + S_w - AP_kC^TS_z^{-1}CP_kA^T}$$

As indicated equation, if we have large noise ($\boldsymbol{S_w}$) we have more error so, we can not trust this estimation. And when $\boldsymbol{S_w}$ is large $\boldsymbol{S_z}$ is also large.

Homework

Do the Kalman Filter Simulation for a vehicle which goes in a straight line.

Measurement noise is 10. Acceleration noise is 0.2

Homework Answer

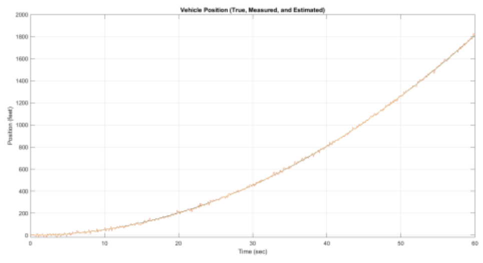
$\boldsymbol{v_{k+1} = v_k + Tu_k + \tilde{v}_k}$ where T is time, $\boldsymbol{\tilde{v}_k}$ is velocity noise.

$\boldsymbol{p_{k+1} = p_k + Tv_k + \frac{1}{2}T^2u_k + \tilde{p}_k}$ where $\boldsymbol{\tilde{p}_k}$ is position noise.

$\boldsymbol{x_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix}}$ where $\boldsymbol{x_k}$ is state vector.

Linear system equation is; $\boldsymbol{x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} u_k + w_k}$

$\boldsymbol{y_k = [1 \quad 0]x_k + z_k}$ where z_k is measurement noise $\boldsymbol{S_w = E(xx^T) = E\left(\begin{bmatrix} p \\ v \end{bmatrix} [p \quad v]\right)}$



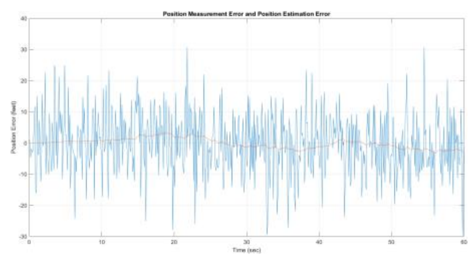
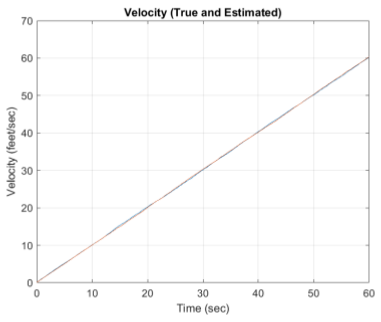


Figure. Plot of Vehicle Position & Vehicle Position Error Comparison



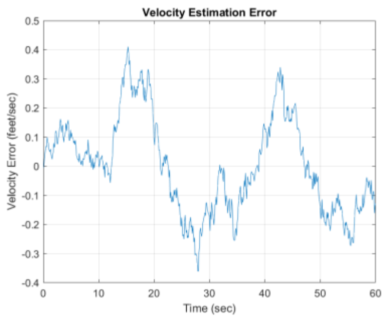


Figure. Plot of Velocity Comparison & Velocity Estimation Error

Week 11

Diagonalization

$A = P\,D\,P^{-1}$ Eigenvectors gives system dynamic and eigenvalue gives the energy in the system.

PCA

To understand system by matrix we need square matrix and in some conditions we have to transform matrix to square matrix. $A_x A_x^T = C_{xx}$

Covariance matrix is always diagonal, symmetric and square.

Note: Unitary Matrix is $p^T = p^{-1}$ $\quad V_n(\theta) = y_n - 2\theta R_{xy} + \theta^T R_{xx}\theta$

With a manipulation (linear algebra),

$$V(\theta) = y - R_{xy}^T R_{xx}^{-1} R_{xy}^T + \left(\theta - R_{xx}^{-1} R_{xy}^N\right)^T R_{xx}^N\left(\theta - R_{xx}^{-1} R_{xy}^N\right)$$

$$R_{xx}^{-1} R_{xy}^N = \hat{\theta} \qquad V_{N,min} = y - R_{xy}^T R_{xx}^{-1} R_{xy}^T$$

Then find eigenvector of R_{xx}^N, $\quad Q_R = \begin{bmatrix} q_{11} & q_{12} & q_{1N} \\ q_{21} & q_{22} & q_{2N} \\ & \dots & \\ q_{N1} & q_{N2} & q_{NN} \end{bmatrix}$

Then find eigenvalue of R_{xx}^N, $\quad \Lambda_R = \begin{bmatrix} \lambda_1 & \dots & 0 \\ 0 & \lambda_2 & 0 \\ 0 & \dots & \lambda_N \end{bmatrix}$

So, $R_{xx}^N = Q_R \Lambda_R Q_R^T \qquad E^T = \left(\theta - \hat{\theta}\right)^T Q_R \qquad V_N(\theta) = V_{N,min}(\theta) + E^T \Lambda_R E$

$V_N(\theta) = V_{N,min}(\theta) + \boxed{\sum_{k=1}^{N} \lambda_k v_k^2} \rightarrow$ The goal is reduce this part

Important Note: If you know x_1 and this gives you an information about x_2, it means there is a correlation between x_1 and x_2.

Kalman Filter

We need standard deviation and mean to estimate in stochastic process. The main idea is sensing with not just a one sensor but multiple to decrease error.

Biased/Unbiased Estimators $y_\alpha(x_1, x_2) = (1-\alpha)x_1 + \alpha x_2 \qquad y(x_1, x_2) = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}x_1 + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}x_2$

$\sigma_y^2 = \frac{1}{\frac{1}{\sigma_1^2}+\frac{1}{\sigma_2^2}} \qquad y(x_1, x_2) = \frac{v_1}{v_1+v_2}x_1 + \frac{v_2}{v_1+v_2}x_2 \qquad v_y = v_1 + v_2$

Fusing scalar estimates $K = \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2} = \frac{v_2}{v_1+v_2} \qquad y(x_1 + x_2) = x_1 + K(x_2 - x_1) \qquad \sigma_y^2 = (1-K)\sigma_1^2$

Homework: Do the PCA example in MATLAB.

Homework Answer:

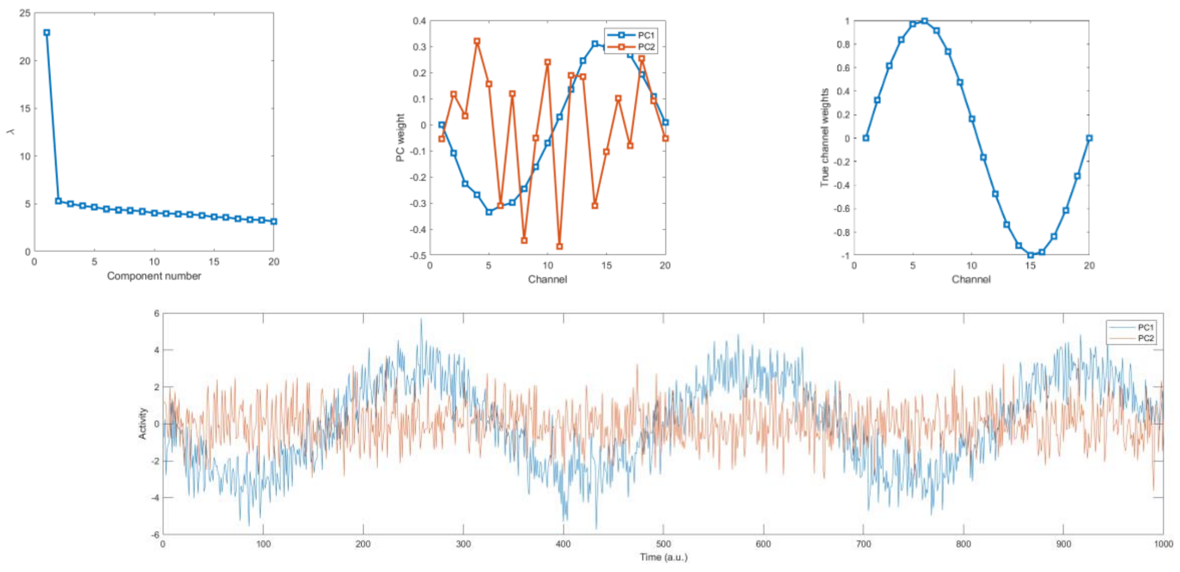


Figure. Plots of the script

Week 12

State-Space Representation

State-space representation is used for a physical system to transform a mathematical model. Relationship between input, output, noise is described by state vectors. When it is applied, differential equations become algebraic equations, matrices. And manipulation or control in algebra is easier than in differentials that's why state-space representation is commonly used in control engineering. Also, there is no information lost in state-space transformation which makes this approach a keystone.

$$\dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{A}(\boldsymbol{t})\boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{B}(\boldsymbol{t})\boldsymbol{u}(\boldsymbol{t}) \qquad \boldsymbol{y}(\boldsymbol{t}) = \boldsymbol{C}(\boldsymbol{t})\boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{D}(\boldsymbol{t})\boldsymbol{u}(\boldsymbol{t})$$

$x(t)$ is state vector	$y(t)$ is output vector
$A(t)$ is state matrix	$C(t)$ is output matrix
$B(t)$ is input matrix	$D(t)$ is feedback matrix

Example by Ljung (DC servo motor):

$$\boldsymbol{u}(\boldsymbol{t}) = \boldsymbol{R_a}\boldsymbol{i}(\boldsymbol{t}) + \boldsymbol{L_a}\frac{\boldsymbol{di}(\boldsymbol{t})}{\boldsymbol{dt}} + \boldsymbol{s}(\boldsymbol{t})$$

$u(t)$ is input (applied voltage) and $s(t)$ is back electromotive force.
$s(t) = k_v \frac{d}{dt} C$ $\quad$ $\eta(t)$ is motor shaft and torque is $\quad$ $T_a(t) = K_a i(t)$

$J \frac{d^2}{dt^2}\eta(t) = T_a(t) - T_l(t) - f\frac{d}{dt}\eta(t)$ J is moment of inertia and f is viscous friction.

In state-space form, $\frac{d}{dt}x(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \beta/\tau \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ \gamma'/\tau \end{bmatrix} T_l(t)$, $\eta(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$

$$x(t) = \begin{bmatrix} \eta(t) \\ \frac{d}{dt}\eta(t) \end{bmatrix} \quad \tau = \frac{JR_a}{fR_a + k_a k_v} \quad \beta = \frac{k_a}{fR_a + k_a k_v} \quad \gamma' = -\frac{R_a}{fR_a + k_a k_v} \quad x(t+T) = A_T(\theta)x(t) + B_T(\theta)u(t)$$

$$\theta = \begin{bmatrix} \tau \\ \beta \end{bmatrix} \quad A_T(\theta) = \begin{bmatrix} 1 & \tau\left(1 - e^{-\frac{T}{\tau}}\right) \\ 0 & e^{-\frac{T}{\tau}} \end{bmatrix} \quad B_T(\theta) = \begin{bmatrix} \beta\left(\tau e^{-\frac{T}{\tau}} - \tau + T\right) \\ \beta\left(1 - e^{-\frac{T}{\tau}}\right) \end{bmatrix} \quad y(t) = \eta(t) + v(t)$$

Process Noise

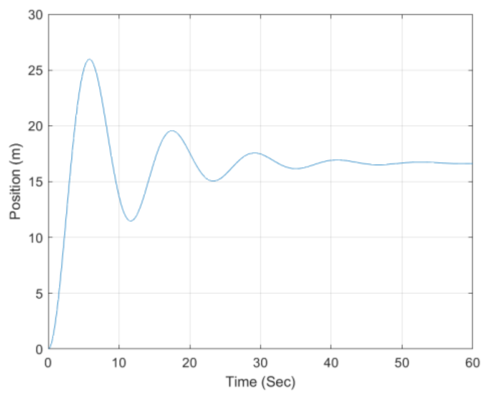
Process noise is the feature of the system changes over time without the information how these changes happen. When the model predicts these changes with stochastic process, after some time noise is created.

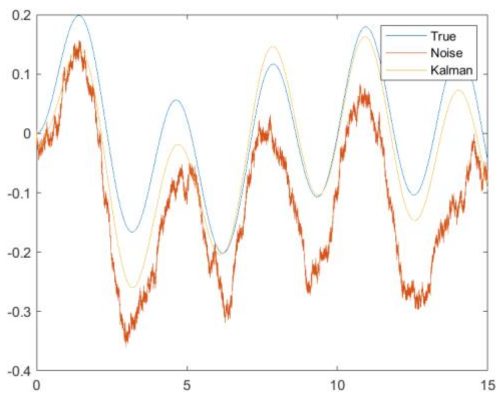
Measurement Noise

Measurement noise comes from the measurement process not the system process and it is estimated by auto-covariance like process noise. After noise is calculated, estimating the next state from past states is the final step.

Homework: Create the Kalman Filter of mass-spring-damper system in MATLAB.

Homework Answer:





Figures. Position of the Vehicle & System Dynamics Comparison

Week 13

State-Space Models

Performing system identification on transfer function models can be happened by state-space forms. State-space models represent a system's dynamics via state equation and output equation. By the benefit of state equation we do not need store past inputs since, it summarizes the effect of all past inputs. $\boldsymbol{x[k+1] = Ax[k] + Bu[k]}$, $\boldsymbol{y[k] = Cx[k] + Du[k]}$

Converting state-space to transfer function can be applied by z-transform.
$\boldsymbol{zX(z) - zx[0] = AX(z) + BU(z)}$, $\boldsymbol{Y(z) = CX(z) + DU(z)}$

$$\boldsymbol{G(z) = \frac{C\, adj(zI-A)B + D\, det(zI-A)}{det(zI-A)}}$$

Poles of the system are where $\boldsymbol{det(zI-A) = 0}$, which are the eigenvalues of A.

Ho-Kalman

To approach state-space realization (problem) Ho-Kalman method is used in common. As Markov parameters are not available to us generally, this method is restrictive but it is useful too.

$$\boldsymbol{x[n] = A^n x[0] + \underset{\substack{\downarrow \\ \text{controllability matrix}}}{[A^{n-1}B]} \begin{bmatrix} u[n-1] \\ u[0] \end{bmatrix}}, \quad \boldsymbol{\begin{bmatrix} y[0] \\ y[n-1] \end{bmatrix} = \underset{\substack{\downarrow \\ \text{observability matrix}}}{\begin{bmatrix} C \\ CA^{n-1} \end{bmatrix}} x[0] + \begin{bmatrix} D & 0 \\ CB & D \end{bmatrix} \begin{bmatrix} u[0] \\ u[n-1] \end{bmatrix}}$$

Step 1: Form Hankel matrix by multiplying C and O matrices.

Step 2: Factor H = OC into O and C components.

Step 3: Use O and C to find A,B and C matrices.

Singular Value Decomposition

Gain properties of A matrix is given by singular value decomposition.

$\boldsymbol{A = U\,\Sigma V^T}$ where U is output singular vector, V is input singular vector.

SVD is much easier than eigensystem calculation,

$$\boldsymbol{A^T A = (U\,\Sigma V^T)^T (U\,\Sigma V^T) = V\Sigma^2 V^T}$$

Since in this method we are dealing vectors, we will know the direction of inputs and outputs too.

Week 14

Stochastic Identification

So far we have not dealing noise in state-space systems. But in every real-life system there is noise so system identification with noise is required to study. Start with writing state equations $\boldsymbol{x[k+1] = Ax[k] + Bu[k] + \omega[k]}$, $\boldsymbol{y[k] = Cx[k] + Du[k] + v[k]}$. In deterministic system we assume noises are white-noise, $\boldsymbol{\omega[k] = v[k] = 0}$

Step 1: Building Statistical Relationships

We must assume noises are white noise, so their expectation is zero. Then, find the Lyapunov equation for state covariance matrix. $\boldsymbol{\Sigma^s = A\Sigma^s A^T + Q}$
Define G to be correlation between future state and present output.
$\boldsymbol{G = A\Sigma^s C^T + S}$. Then define covariance of outputs. $\boldsymbol{\Lambda_i = \mathrm{E}[y[k+i]y^T[k]]}$
$\boldsymbol{\Lambda_0 = C\Sigma^s C^T + R,} \quad \boldsymbol{\Lambda_i = CA^{i-1}G}$. This means that output covariances can be considered as Markov parameters. For the notation of next steps identify $\boldsymbol{\varsigma_i^d = [A^{i-1}B, \dots]}$ & $\boldsymbol{\varsigma_i^c = [A^{i-1}G, \dots]}$ & $\boldsymbol{C_i = \begin{bmatrix} \Lambda_i & \Lambda_{i-1} \\ \Lambda_{i+1} & \Lambda_i \end{bmatrix}}$ & $\boldsymbol{L_i = \begin{bmatrix} \Lambda_0 & \Lambda_{-1} \\ \Lambda_1 & \Lambda_0 \end{bmatrix}}$

Step 2: Kalman-filter covariance

Kalman filter is a recursive algorithm to estimate future state of the system.
$\boldsymbol{\hat{x}[k] = A\hat{x}[k-1] + K[k-1](y[k-1] - C\hat{x}[k-1])}$, $\quad \boldsymbol{K[k-1] = (G - AP[k-1]C^T)(\Lambda_0 - CP[k-1]C^T)^{-1}}$, $\quad \boldsymbol{P[k] = \mathrm{E}\left[\hat{x}[k]\,\hat{x}^T[k]\right]}$

$\boldsymbol{\hat{x}[p] = \varsigma_p^c\, L_p^{-1} \begin{bmatrix} y[0] \\ y[p-1] \end{bmatrix}}$ & $\boldsymbol{P[p] = \varsigma_p^c\, L_p^{-1}\left(\varsigma_p^c\right)^T}$

Step 3: Kalman-filter state

Show state relationships, $\boldsymbol{\hat{x}[p+1] = A\hat{x}[p] + K[p](y[p] - C\hat{x}[p])}$
$\boldsymbol{\hat{x}[i+q] = \varsigma_p^c\, L_p^{-1} \begin{bmatrix} y[q] \\ y[i+q-1] \end{bmatrix}}$

Step 4: Geometric properties of stochastic systems

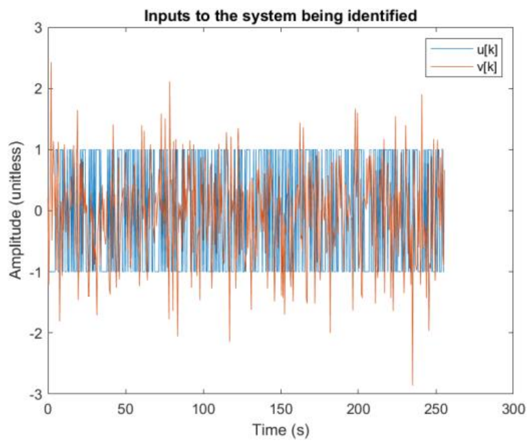
Now assume that noises are not zero, $\boldsymbol{\xi_i = C_i L_i^{-1} Y_p = O_i \hat{X}_i}$. After SVD we can conclude that, row of the future states $\hat{X}_i$ can be found by orthogonally projecting row space of the future output Y_f onto the row space of the past outputs Y_p.

Step 5: Computing the system matrices

Now we can calculate state future states using only output data. $\begin{bmatrix} \hat{X}_{i+1} \\ Y_{i/i} \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix} \hat{X}_i + \begin{bmatrix} \rho_\omega \\ \rho_v \end{bmatrix}$. After solving for residuals and noise covariance matrices we get, $\boldsymbol{P[i+1] = AP[i]A^T + \hat{Q}_i}$

FINAL EXAM

1. Estimation of a Discrete-Time System with 2 Delays,Zeros & 3 Poles



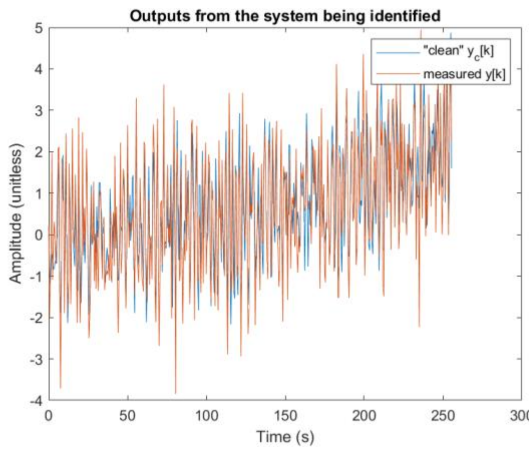
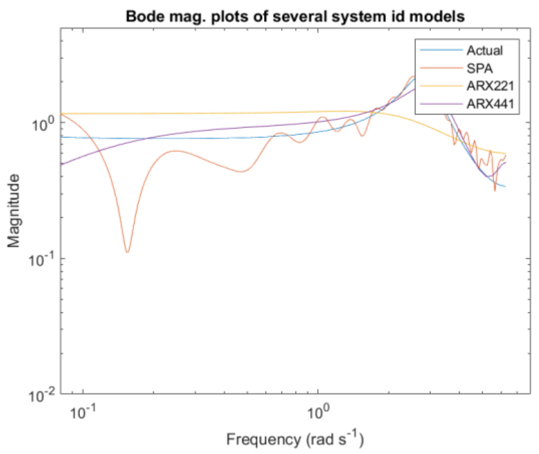
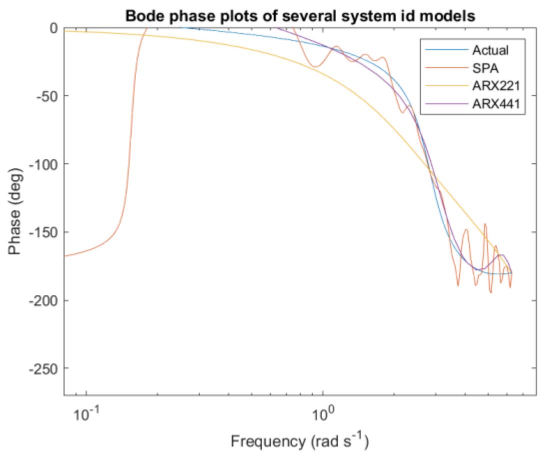
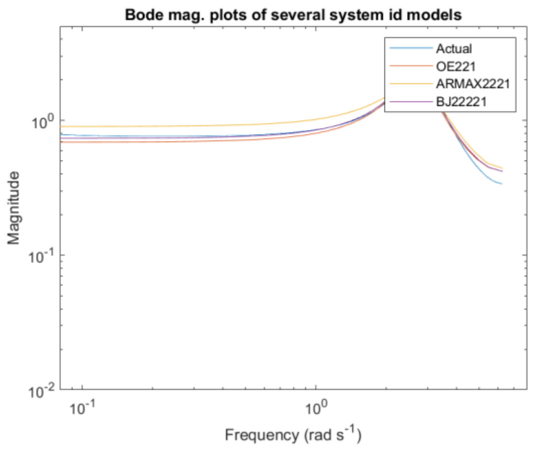


Figure 1. Inputs and Outputs of the System

Comment: In left side, blue line represent input signal which is a binary and orange line represents sensor noise (random). In right side, blue line is cleaned output signal and orange line is noisy output data.







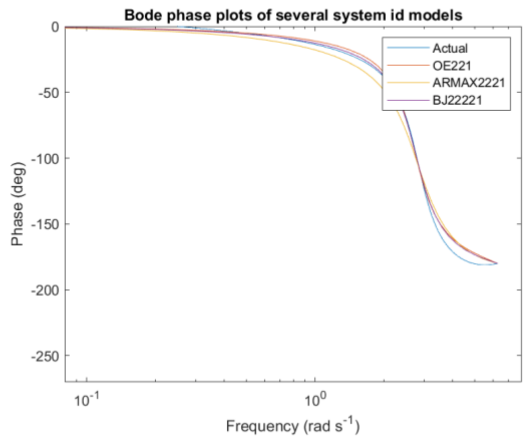
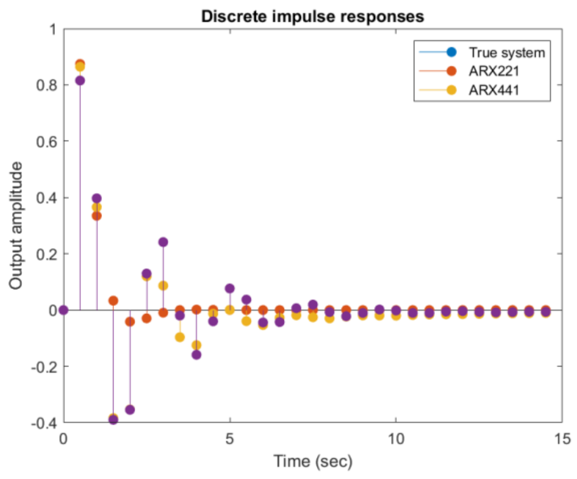


Figure 2. Bode Plot of Several Models

Comment: At the first line ARX441 is the closest to the actual system and SPA is also close but not low frequencies. At the second line, OE221 and BJ22221 is perfectly fit and ARMAX2221 is close to fit to the real system. Despite complexity of ARX221 and OE221 are same the result is divergent. However ARX model is preferred because of their simplicity of optimization, it is not well- fitted to the actual system as other models.



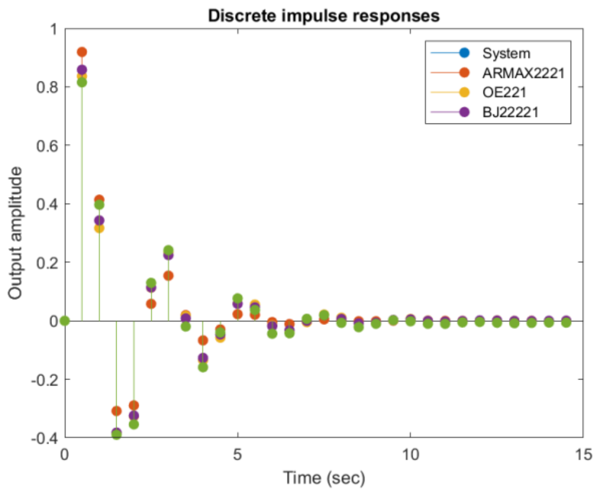


Figure 3. Impulse Responses of Several Models

Comment: As expected from bode plot, ARX models are not close to the impulse response of the real system but, other models fit well especially OE221. And this also gives that the amount of knowledge we have does not mean we will know the system itself better because, OE221 has less coefficient than other but it's results are more closer to the system.

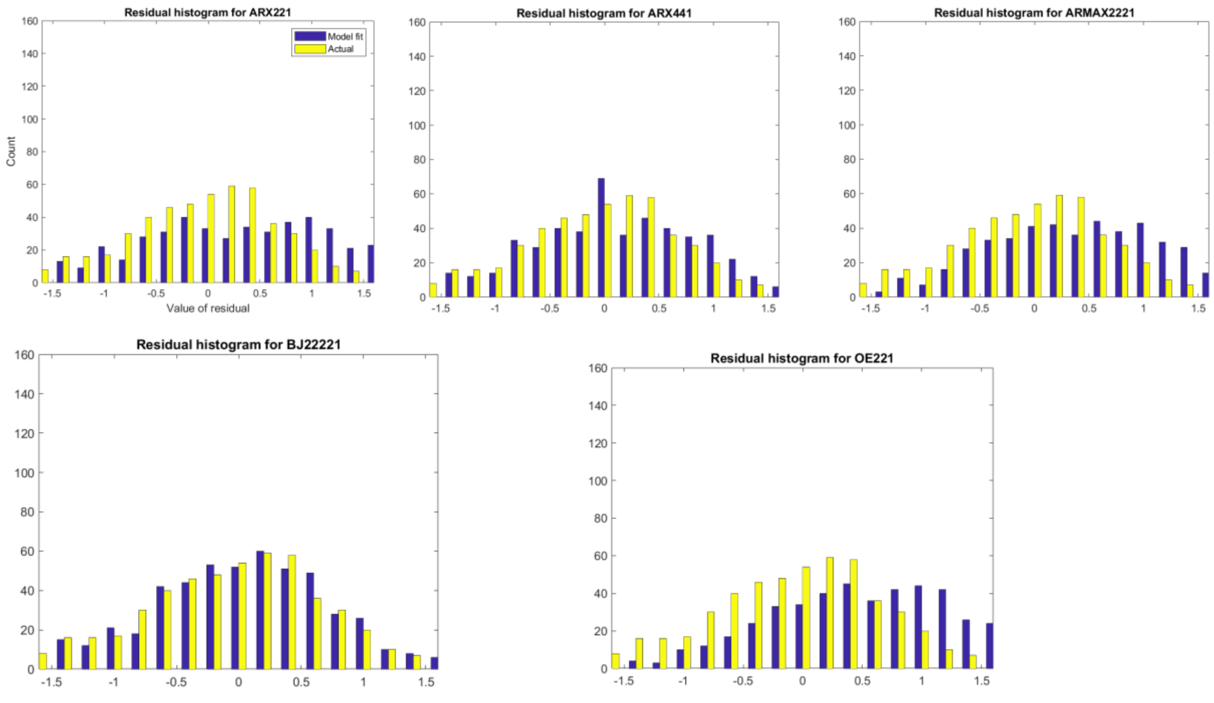


Figure 4. Residual Histograms for Several Models

Comment: For the residuals BJ2221 fits well but other models have failed.

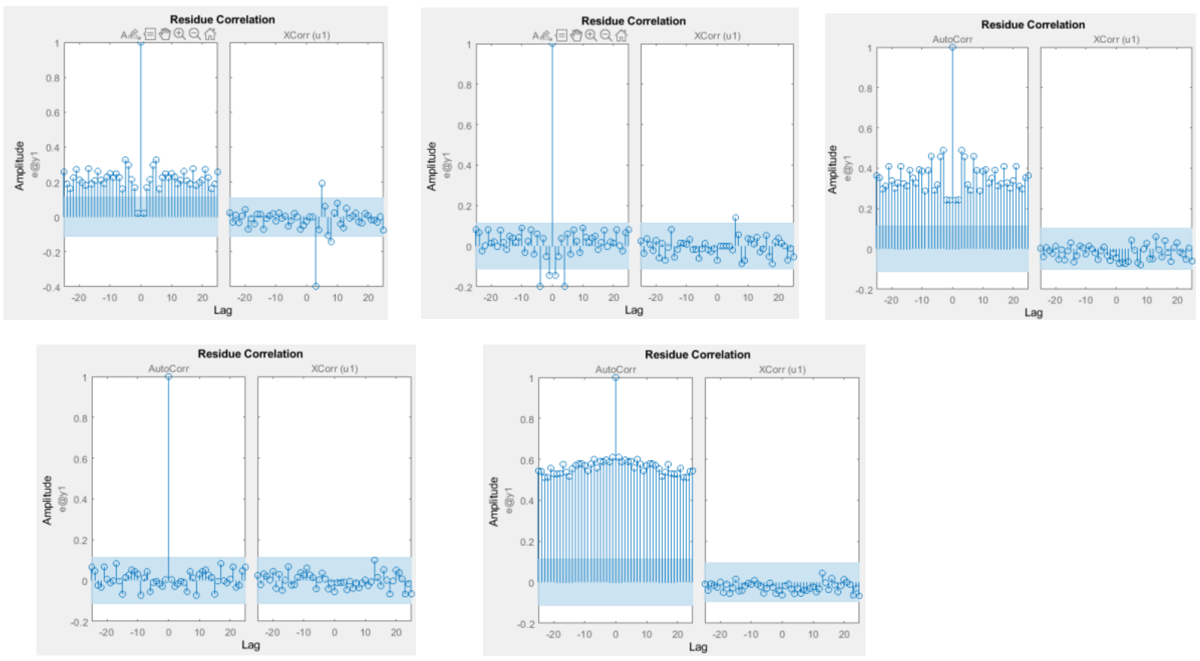


Figure 5. Residue Correlations of Several Models

Comment: If the model fits in the blue box, it means model is works well in autocorrelation perspective. So, again BJ2221 has won the competition.

2. Estimation of the System with Initial Model Structure Selection

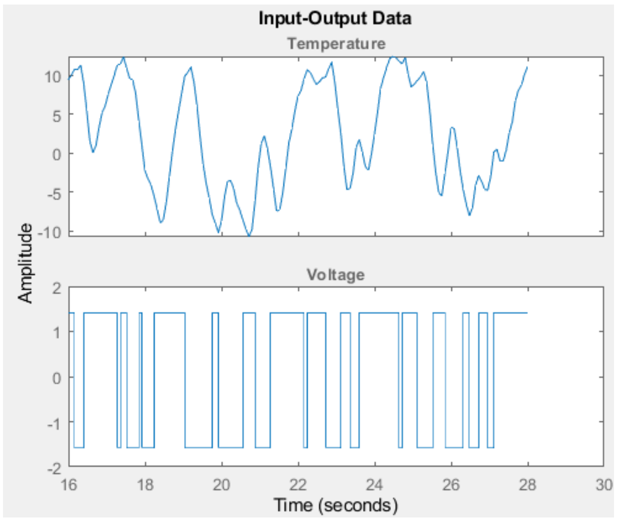


Figure 1. Input & Output Data

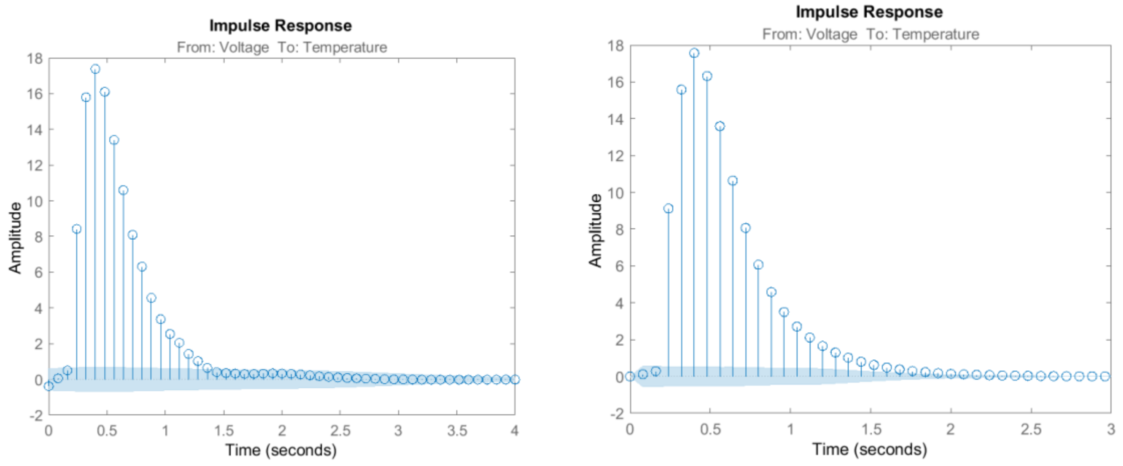


Figure 2. Impulse Response

Comment: The filled light-blue region shows the confidence interval for the insignificant response in this estimation. There is a clear indication that the impulse response leaves the uncertainty region after 3 samples. This points to a delay of three intervals.

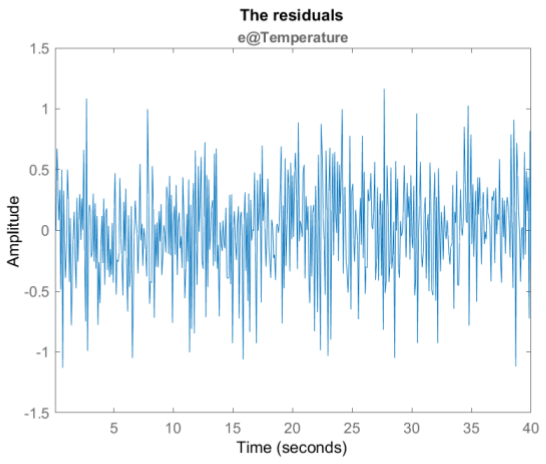


Figure 3. Residuals

Comment: We see that the residuals are quite small compared to the signal level of the output, that they are reasonably well uncorrelated with the input and among themselves. We can thus be satisfied with the model th2.

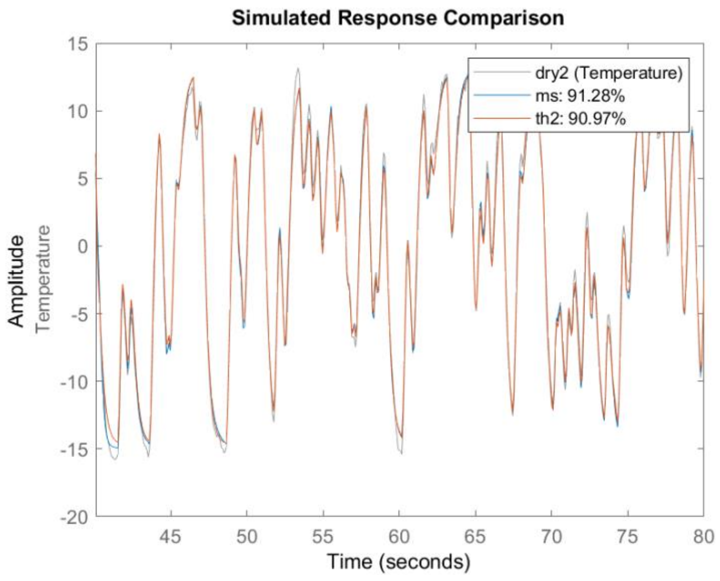


Figure 4. Simulated Response Comparison

Comment: The comparison plot indicates that the two models are practically identical.

3. Kalman Filtering on Mechanical System with 2 Mass-Damper & 3 Spring

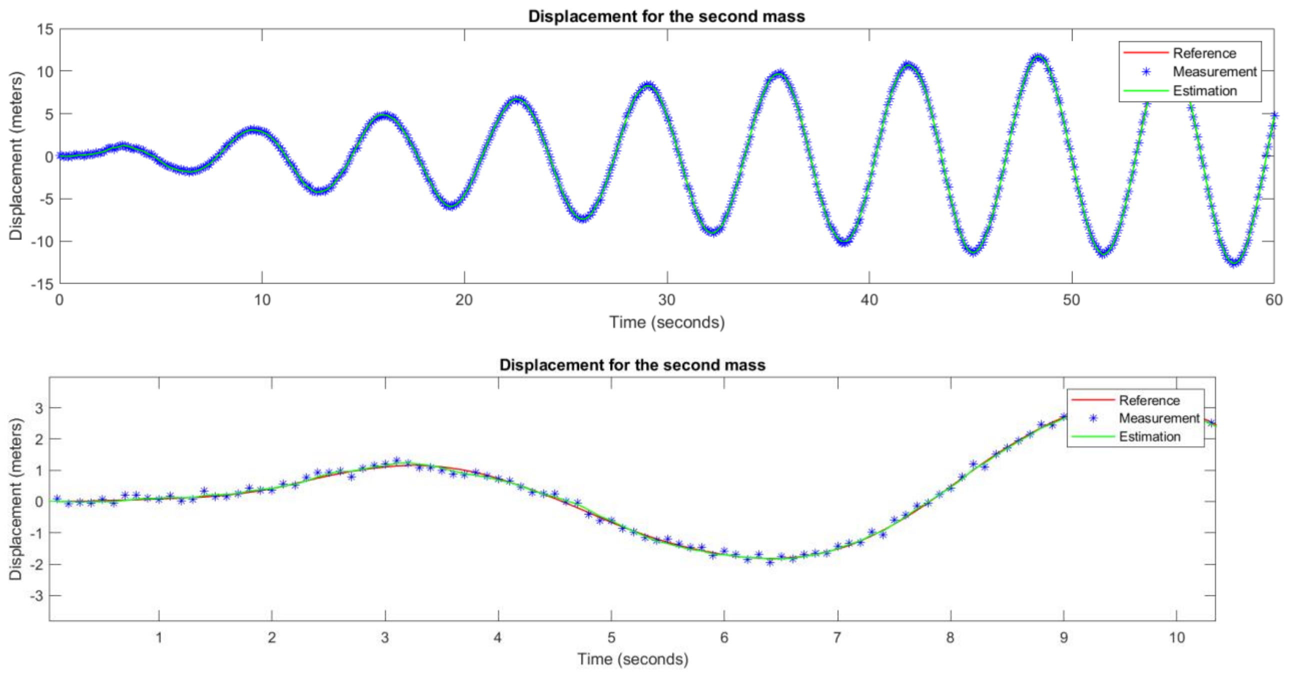


Figure 1. Displacement of the Second Mass

Comment: As expected displacement is sinusoidal like the force applied. When we zoom it, we can see measurement and estimation are identical. So the Kalman filter works well.

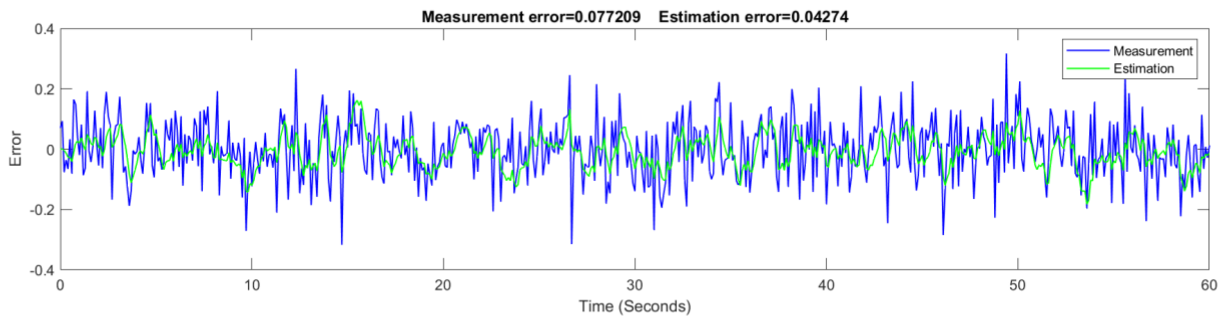


Figure 2. Measurement Error vs. Estimation Error