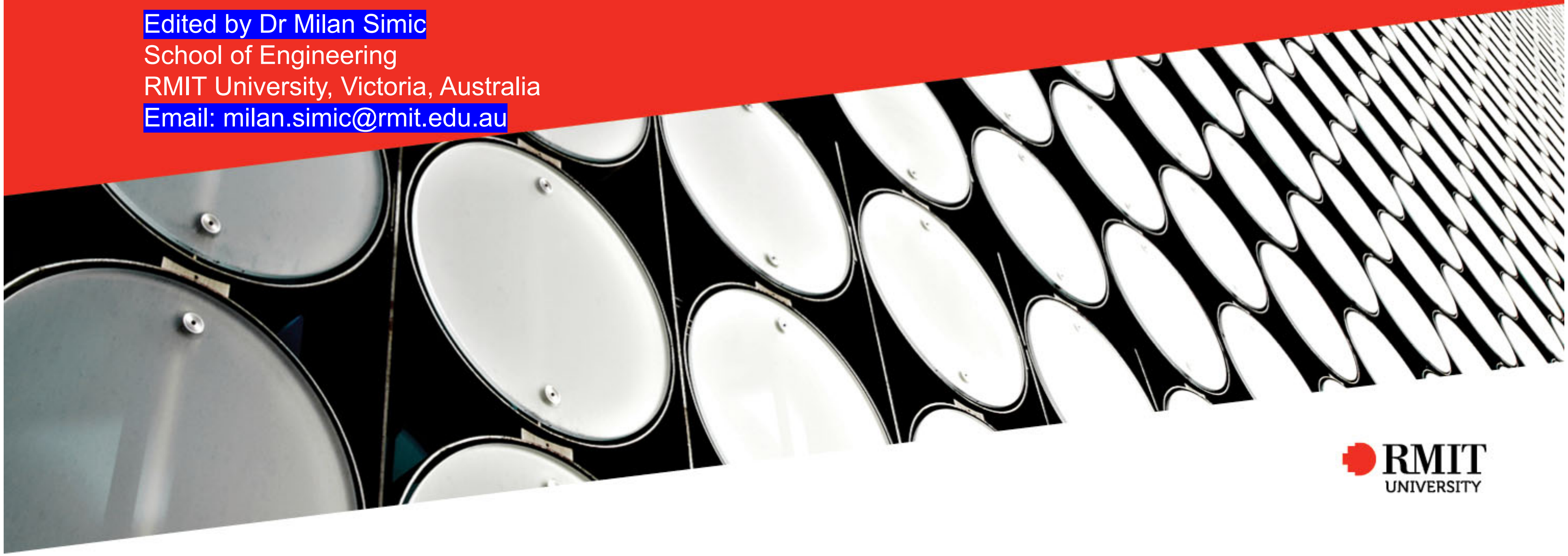
Week 10 – Modelling and System Identification

Advanced Mechatronics System Design – MANU2451

Dr Chow Yin LAI

Edited by Dr Milan Simic
School of Engineering
RMIT University, Victoria, Australia
Email: milan.simic@rmit.edu.au



New Teaching Schedule

Week		Class Activity Before	Lecture	Class Activity During or After
1			Introduction to the Course / Introduction to LabVIEW	LabVIEW Programming
2			Introduction to LabVIEW / Data Acquisition	LabVIEW Programming
3			Gripper / Introduction to Solidworks / Safety	Gripper Design
4			Sensors I	myRIO Programming for Sensor Signal Reading / Gripper Design
5			Sensors II	myRIO Programming for Sensor Signal Reading
6			Actuators I	LabVIEW Tutorial
7		LabVIEW Assessment.	DC Motors I	Matlab Simulink Simulation
8		Design report submission	DC Motors II	Matlab Simulink Simulation / myRIO Programming for Control
9			Actuators II	Matlab Simulink Simulation
10			Modeling and System Identification	Matlab Simulink Simulation LabVIEW Simulation
11			Artificial Intelligence I	Matlab Simulation LabVIEW Simulation
12		Gripper Simulation / Submission of Report	Artificial Intelligent II	Revision

Assessment

- Assessment 1 – Well done
- Assessment 2 –
- Assignment 3 and 4 due date 27th of may / or one more week
- Practice Final Quiz week 13
- Assessment 5 Final Quiz week 14

Contents

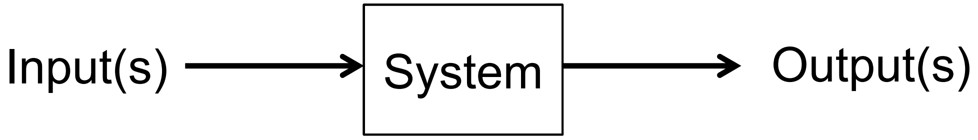
- Introduction
- Identification of Continuous-Time Models in Time Domain
  - Grey Box
  - Black Box
- Identification of Discrete-Time Models
  - Grey Box
  - Black Box
- Identification of Continuous-Time Models in Frequency Domain

Contents

- Introduction
- Identification of Continuous-Time Models in Time Domain
  - Grey Box
  - Black Box
- Identification of Discrete-Time Models
  - Grey Box
  - Black Box
- Identification of Continuous-Time Models in Frequency Domain

System

- System is a complex entity
- Systems exist in Engineering, Business, Bio-systems…
- Set of connected parts, subsystems or elements
- An engineering system is an entity which responds to external input(s) and produces output(s) in response to the input(s) and its state.



- E.g. if a voltage (input) is supplied to a motor, then the rotation of the shaft is the output.
- If force (input) is applied on a piece of wooden block, then the displacement of the wooden block is the output.

System Modelling

- System Modelling Overview
- Block Diagrams and State-Space Modelling
- Electro-Mechanical Systems Modelling
- Computer Aided Modelling

System Modelling Overview

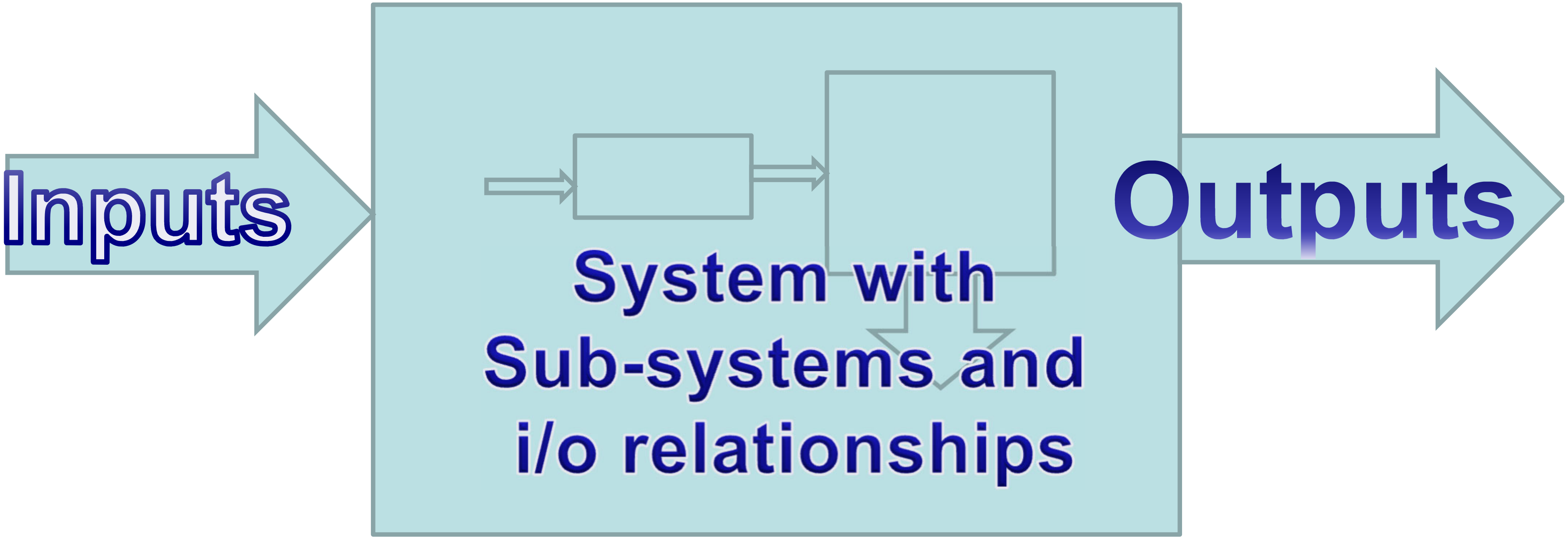
- Systems,
  - Mechatronics, Physical Networks, Analogue vs. Discrete Systems
- Model is a mathematical representation of the system
- Control systems' models have
  - inputs (sensors) and
  - Control Unit
  - outputs (actuators)
- They combine mechanical and electrical engineering

Q: *Electrical vs. Electronics*

System Modelling Overview

- System is a closed set of elements (entities, or items) and defined relationships among them.
- A model is an idealised representation of the system.
- Model has its properties well-defined.
- Excitations (inputs) are applied to the system, and responses (outputs) are measured from the actual system.
- *Each element of a mechatronics system possess causal, i.e.* ***Input / Output*** *relationship.*

System Modelling Overview

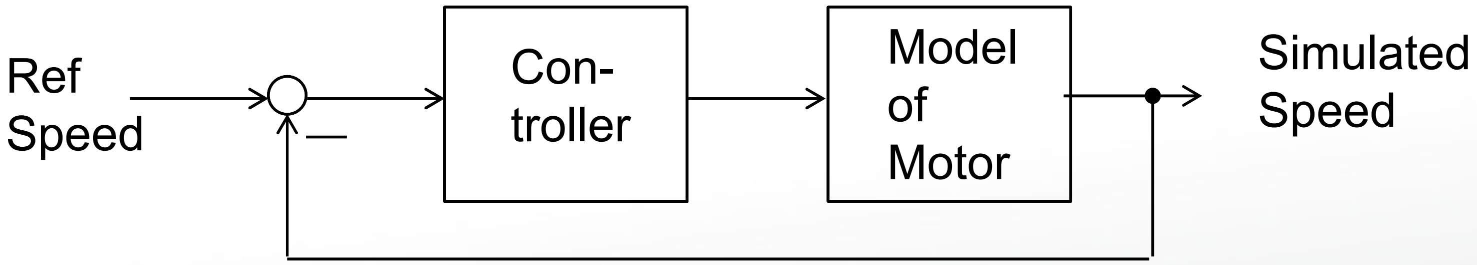


Models

- Physical Model (Prototype)
- Analytical
  - State space model, linear graph, bond graph, block diagram, signal-flow graph, transfer function model, frequency domain model
- Computer
- Experimental / Model based design
  - Apply inputs and measure outputs

Modelling

- Why modelling?
  - It allows us to understand the dynamic characteristic of a system so that we can:
    - Re-design the system for better performance (e.g. higher resonance frequency, lower resonance amplitude)
    - Design and simulate a controller



– These models allow you to design and simulate the system in computer, before you implement it on the real system.

» Shorten system development time, For safety reason

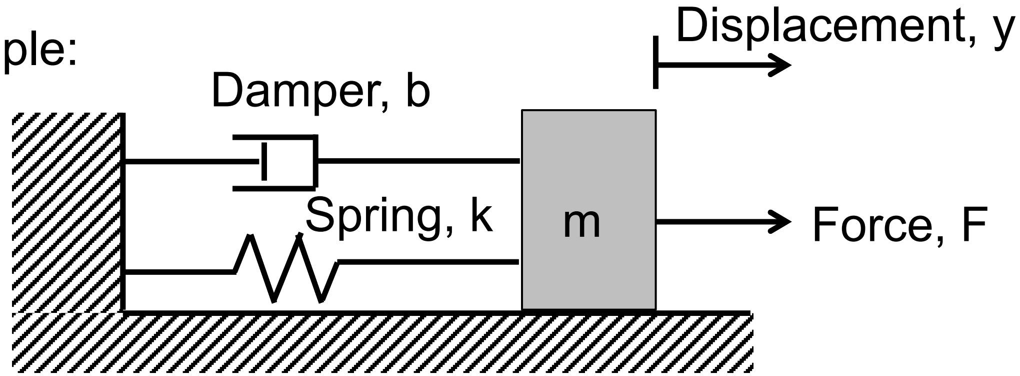
» For cost reason

White Box

- Of course, if we want to use the model for simulation and control design, the model needs to provide a response which is similar to the actual system.
- How do we find the model of a system?
  - 1) Physical modeling (White Box):
    - If the structure of the system is largely known, we can produce a block diagram of the system and its functional components.
    - Derive model of each functional component using any known physics law (e.g. Newton's Law, Hooke's Law)
    - Determine the behaviour using the interconnection of components.

White Box

- Example:



- Newton's law: $F(t) = ma = m\dfrac{d^2y}{dt^2}(t)$
- Force exerted by spring is proportional to displacement $F(t) = ky(t)$
- Force exerted by damper is proportional to velocity $F(t) = b\dfrac{dy}{dt}(t)$
- Therefore: $m\dfrac{d^2y}{dt^2}(t) = F(t) - ky(t) - b\dfrac{dy}{dt}(t)$ ⇨ $\underbrace{m\dfrac{d^2y}{dt^2}(t) + b\dfrac{dy}{dt}(t) + ky(t)}_{\text{Output Dynamics}} = \underbrace{F(t)}_{\text{Input}}$

- The exact values of m, b, k are read off from spec sheets, or by physical measurement.

Grey Box

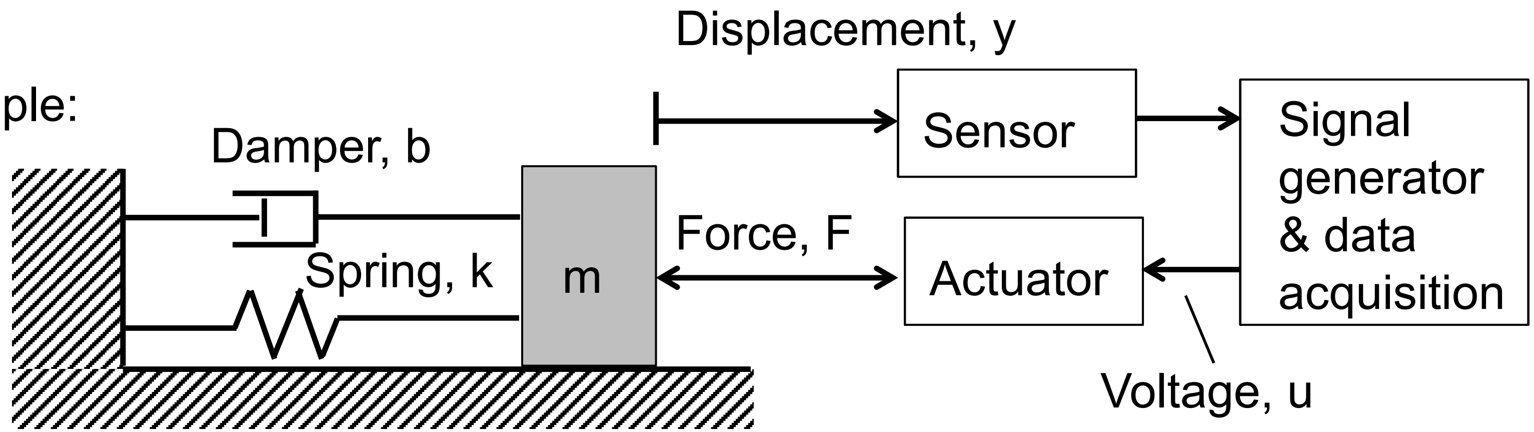
- 2) Parameter identification (Grey Box):
  - Form of model is known, but parameters unknown.
    - E.g. we know that the model looks like:

$$\underbrace{m\frac{d^2y}{dt^2}(t)+b\frac{dy}{dt}(t)+ky(t)}_{\text{Output Dynamics}}=\underbrace{F(t)}_{\text{Input}}$$

    - But we do not know the values of *m, b, k.*
  - Apply external excitation and observe response.
  - Find parameters of the model such that observed response matches the output of the model.

Grey Box

- Example:



Displacement, y

Sensor

Signal generator & data acquisition

Damper, b

Spring, k

m

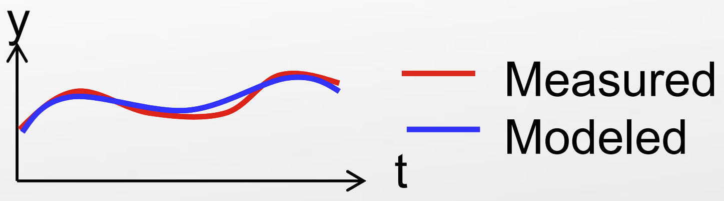
Force, F

Actuator

Voltage, u

- Use a signal generator and actuator to apply varying force to the mass.
- Observe the displacement based on the input force.
- Collect data of both the applied force and the displacement.
- Determine parameters of the model which allows the response of model to match closely with the measured response.

$$\underbrace{m\frac{d^2y}{dt^2}(t)+b\frac{dy}{dt}(t)+ky(t)}_{\text{Output Dynamics}}=\underbrace{F(t)}_{\text{Input}}$$



Black Box

System identification (Black Box):

Form of model and parameter unknown.

Apply external excitation and observe response.

Determine type of mathematical formula that relates excitation and response.

Find parameters of the model such that observed response matches the output of the model.

Enigma machine sells for a record $463,500
https://newatlas.com/enigma-auction-record/46841/

RMIT University

Implant Testing Project

***Non-destructive** test methods and their application to measure the stability and osseointegration of bone anchored endosseous implants*

Measuring electrical impedance Ze instead of mechanical Zm

Measure frequency response of a system using frequency sweep

RMIT University, SoE
in collaboration with
University of Melbourne, Melbourne Dental School

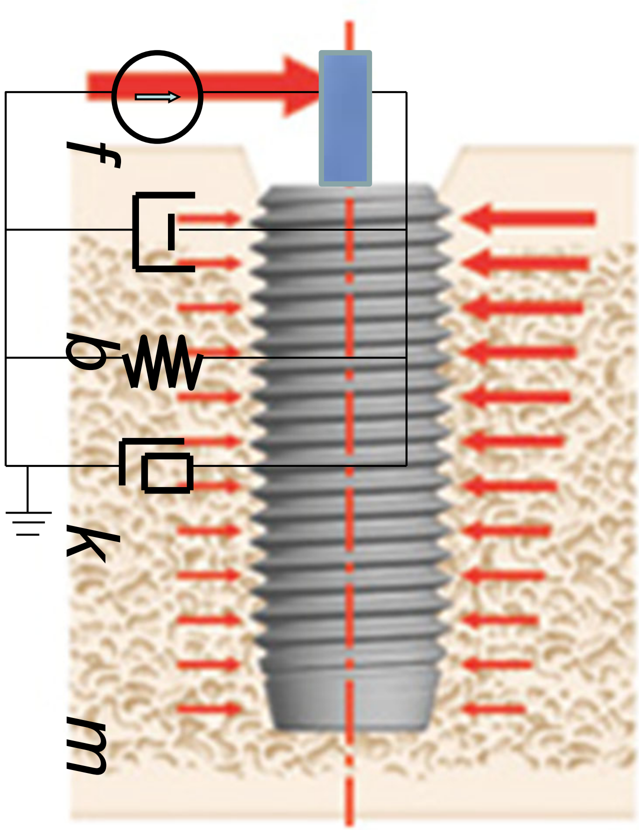
Dr Milan Simic, RMIT University

Dr Dragan Grubor, University of Melbourne

Research Objective:
*Demonstrate the feasibility of using bioimpedance measurements to determine successful **osseointegration.***

Milan.simic@rmit.edu.au

Distributed **Mechanical** Impedance Along the Contact Surface



$$bv + m\frac{dv}{dt} + k\int vdt = f$$

$$w_o = \sqrt{\frac{k}{m}}$$ Undamped Natural Resonance Frequency

b = Damping;

m = Mass (Inertia)

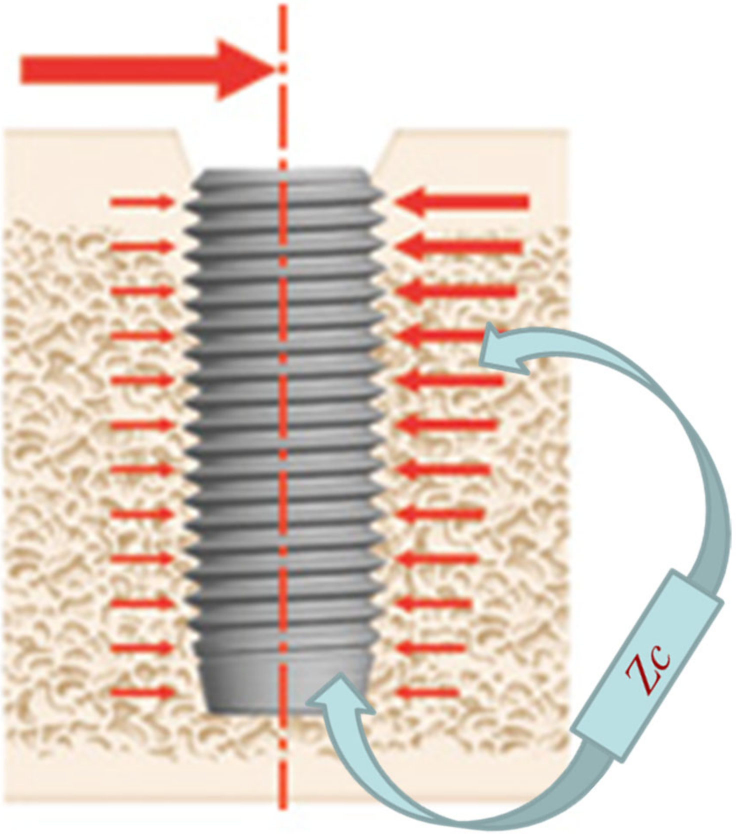
k = Spring (Stiffness)

f = Force, v = Velocity

Osseointegration is an artificial implant structural connection with living bone.

"the formation of a direct interface between an implant and bone, without intervening soft tissue".

Distributed Impedance Along the Contact Surface



$$Z_c = R + j2\pi fL + \frac{1}{j2\pi fC}$$

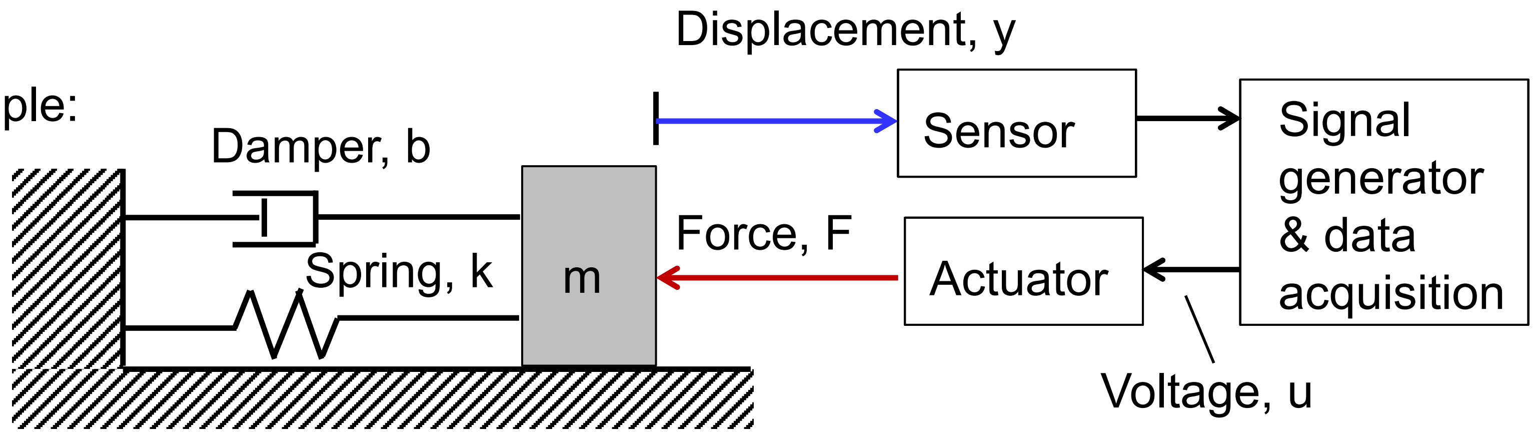
Perform scanning through various frequency bands and measure electrical impedance

Osseointegration is an artificial implant structural connection with living bone.

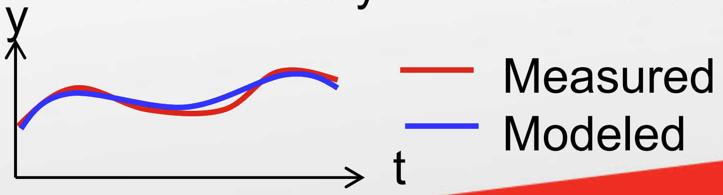
"the formation of a direct interface between an implant and bone, without intervening soft tissue".

Black Box

- Example:



- Use a signal generator and actuator to apply <u>varying</u> force to the mass.
- Observe the displacement based on the input force.
- Collect data of both the applied force and the displacement.
- Analyze collected data to choose a model structure.
  - E.g. $y(k)=a_1y(k-1)+a_2y(k-2)+b_1u(k-1)+b_2u(k-2)$
- Determine parameters of the model which allows the response of model to match closely with the measured response.



Grey Box and Black Box

- In the rest of this lecture, we will focus only on:
  - Grey Box (Parameter Identification)
  - Black Box (System Identification)

Contents

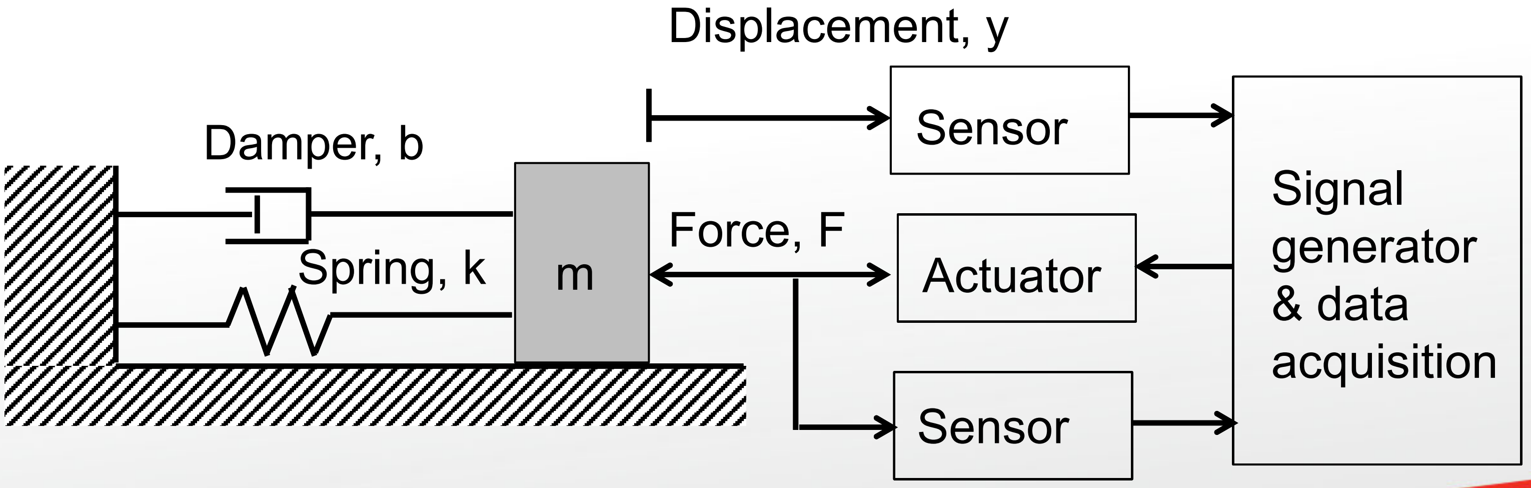
- Introduction
- Identification of Continuous-Time Models in Time Domain
    - Grey Box
    - Black Box
- Identification of Discrete-Time Models
    - Grey Box
    - Black Box
- Identification of Continuous-Time Models in Frequency Domain

Grey-Box Modelling

- Let's consider the mass-spring-damper system earlier.
- The physical model is known, however the parameters are unknown:

$$m\frac{d^2y}{dt^2}(t)+b\frac{dy}{dt}(t)+ky(t)=F(t) \quad \text{or} \quad m\ddot{y}(t)+b\dot{y}(t)+ky(t)=F(t)$$

- Instead of measuring the mass, damping coefficient and spring coefficient individually, we would like to estimate them from input output measurements.



Grey-Box Modelling

- Use a signal generator and actuator to apply varying force to the mass.
- Collect data of both the applied force, the displacement, velocity and acceleration at a constant sampling interval (T).
- Put the measured data in the following form:

$$\underbrace{\begin{bmatrix} \ddot{y}(0) & \dot{y}(0) & y(0) \\ \ddot{y}(T) & \dot{y}(T) & y(T) \\ \ddot{y}(2T) & \dot{y}(2T) & y(2T) \\ \vdots & \vdots & \vdots \\ \ddot{y}(N) & \dot{y}(N) & y(N) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} m \\ b \\ k \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} F(0) \\ F(T) \\ F(T) \\ \vdots \\ F(T) \end{bmatrix}}_{B}$$

- Our next step is to calculate the parameter vector θ. But how?

Grey-Box Modelling

- In previous slide, θ is the true parameters of the system, such that $A\theta = B$.
- In practice, we will not get the exact value of θ.
- What we want is to get an estimate of θ, which we denote as $\hat{\theta}$, such that:
  - $A\hat{\theta} = \hat{B}$ and $\hat{B}$ similar to B
- How do we measure the "similarity" between $\hat{B}$ and B?
- Define the modelling error: $e(k) = B_k - \hat{B}_k$
- Then the error vector for the acquired data would be:

$$E = \begin{bmatrix} e(0) \\ e(T) \\ e(2T) \\ \vdots \\ e(N) \end{bmatrix} = \begin{bmatrix} B_0 - \hat{B}_0 \\ B_T - \hat{B}_T \\ B_{2T} - \hat{B}_{2T} \\ \vdots \\ B_N - \hat{B}_N \end{bmatrix} = \begin{bmatrix} F(0) - \hat{F}(0) \\ F(T) - \hat{F}(T) \\ F(2T) - \hat{F}(2T) \\ \vdots \\ F(N) - \hat{F}(N) \end{bmatrix}$$

Grey-Box Modelling

- For $\hat{B}$ to be similar to B would mean that the modelling error vector E is small.
- The “smallness” of the vector E can be measured using the norm of vectors:

$$
\begin{aligned}
V &= \frac{1}{2}\left(e^2(0)+e^2(T)+e^2(2T)+\cdots+e^2(N)\right)\\
&= \frac{1}{2}\sum_{k=0}^{N} e^2(k)\\
&= \frac{1}{2}\sum_{k=0}^{N} \left(B_k-\hat{B}_k\right)^2
\end{aligned}
$$

- Which can also be written in the vector form:

$$
\begin{aligned}
V &= \frac{1}{2}\left(B-\hat{B}\right)^T\left(B-\hat{B}\right)\\
&= \frac{1}{2}\left(B-A\hat{\theta}\right)^T\left(B-A\hat{\theta}\right)
\end{aligned}
$$

Grey-Box Modelling

- It is repeated here:

$$V = \frac{1}{2}\left(B - \hat{B}\right)^{T}\left(B - \hat{B}\right)$$
$$= \frac{1}{2}\left(B - A\hat{\theta}\right)^{T}\left(B - A\hat{\theta}\right)$$

- If we see the above as a “cost function”, and if we minimize this cost function, we will be able to obtain an estimate of $\hat{B}$ which is close to B.
- To minimize this cost function, note that:

$$V = \frac{1}{2}\left(B - A\hat{\theta}\right)^{T}\left(B - A\hat{\theta}\right) = \frac{1}{2}\left[B^{T}B - \underbrace{B^{T}A\hat{\theta}}_{\text{scalar}} - \underbrace{\hat{\theta}^{T}A^{T}B}_{\text{scalar}} + \hat{\theta}^{T}A^{T}A\hat{\theta}\right] = \frac{1}{2}\left[B^{T}B - 2\hat{\theta}^{T}A^{T}B + \hat{\theta}^{T}A^{T}A\hat{\theta}\right]$$

- Minimization means finding the derivative and setting this to zero:

$$\frac{dV}{d\hat{\theta}} = \frac{1}{2}\left[-2A^{T}B + 2A^{T}A\hat{\theta}\right] = 0$$
$$A^{T}A\hat{\theta} = A^{T}B$$
$$\hat{\theta} = \left(A^{T}A\right)^{-1}A^{T}B$$

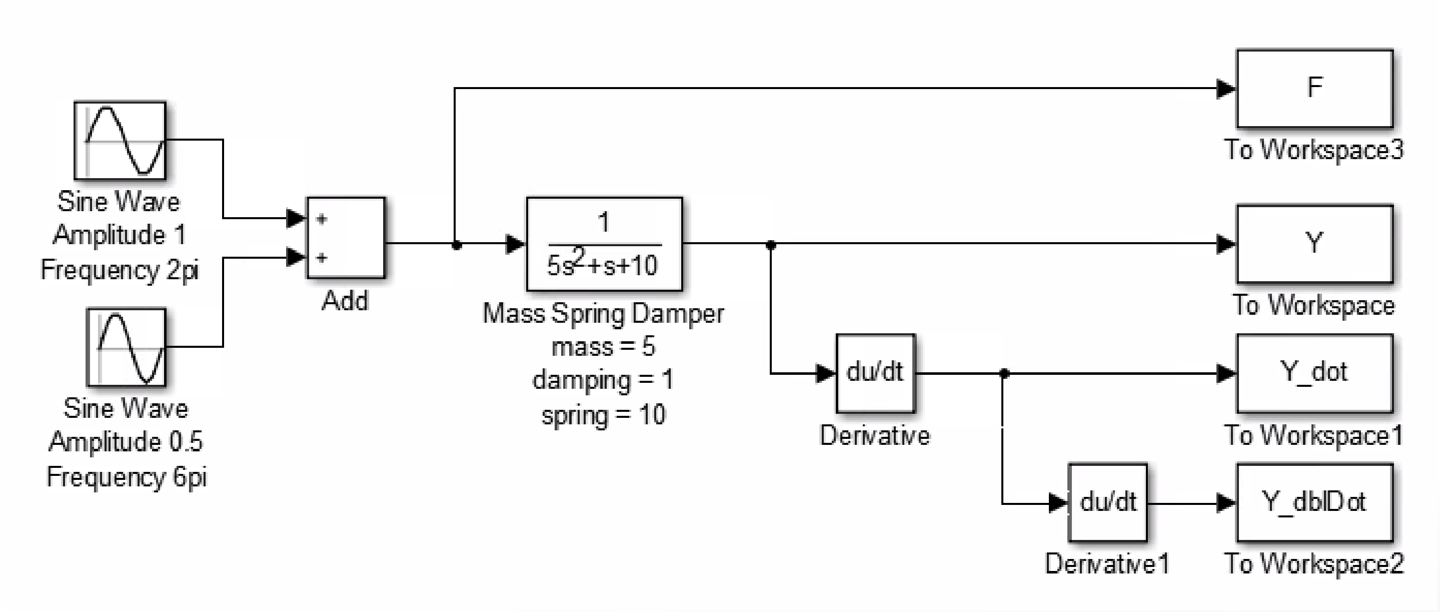
Least square estimates of parameters!

Least Squares Estimation

- The Least Squares Estimator has several attractive features for purposes of identification:
  - Large errors are heavily penalized.
  - LS estimate can be obtained by straightforward matrix algebra.
  - The LS criterion is related to statistical variance; properties of the solution can be analyzed according to statistical criteria.
    - $E\{\hat{\theta}\} = \theta$ provided that $E\{A^T e\} = 0$ i.e. The LS estimate is unbiased when disturbance is uncorrelated with the regressor.
      - If experiment is repeated several times to find parameters, then the mean of estimates from different trials goes to the true values of estimates.

MATLAB / Simulink Example

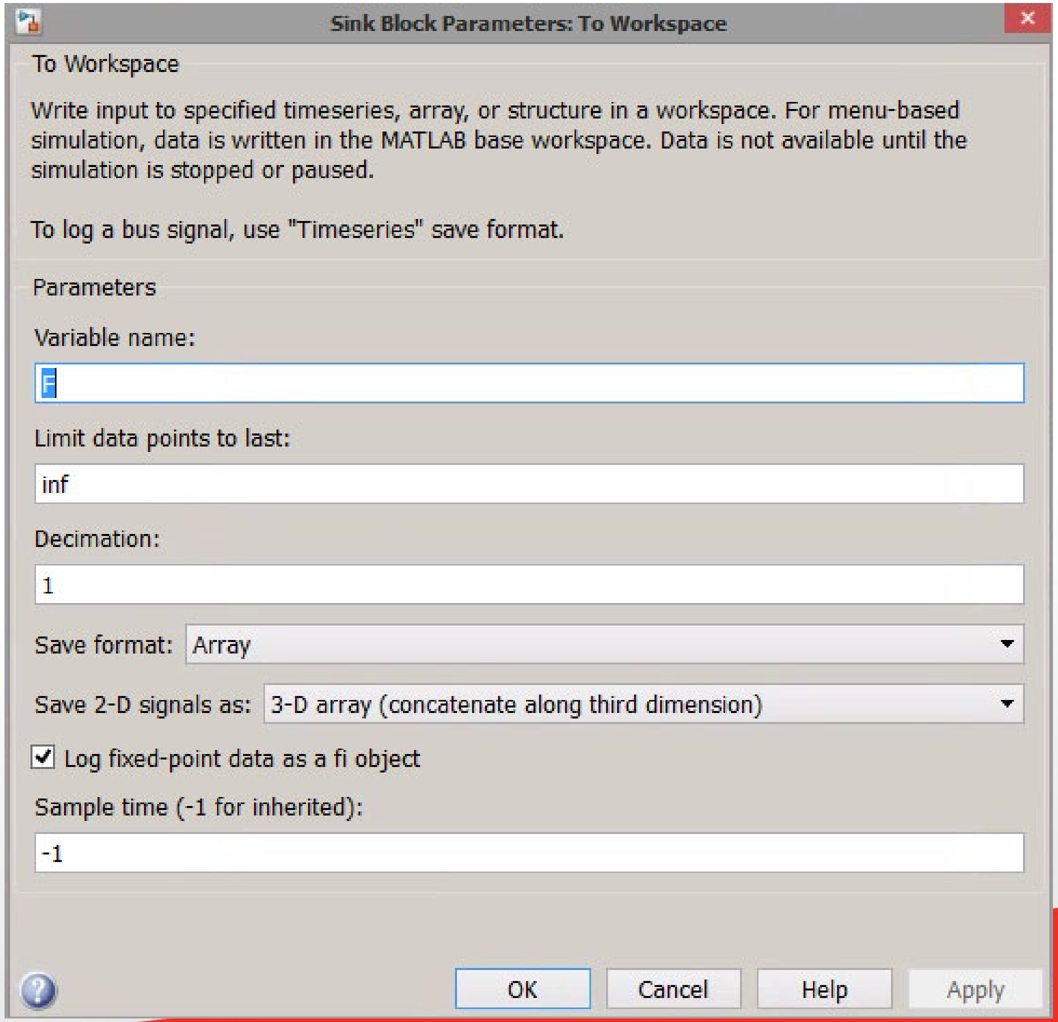
- Create the following Simulink block diagram:



- Of course, we imagine that we do not know the real system, even though for simulation purpose we have to create one with parameters.
- The above diagram shows the collection of input-output data.
- In real experiments, you could use LabVIEW to acquire the data.

MATLAB / Simulink Example

- The “to Workspace” blocks should be formatted as follows:



Sink Block Parameters: To Workspace

To Workspace

Write input to specified timeseries, array, or structure in a workspace. For menu-based simulation, data is written in the MATLAB base workspace. Data is not available until the simulation is stopped or paused.

To log a bus signal, use "Timeseries" save format.

Parameters

Variable name: F

Limit data points to last: inf

Decimation: 1

Save format: Array

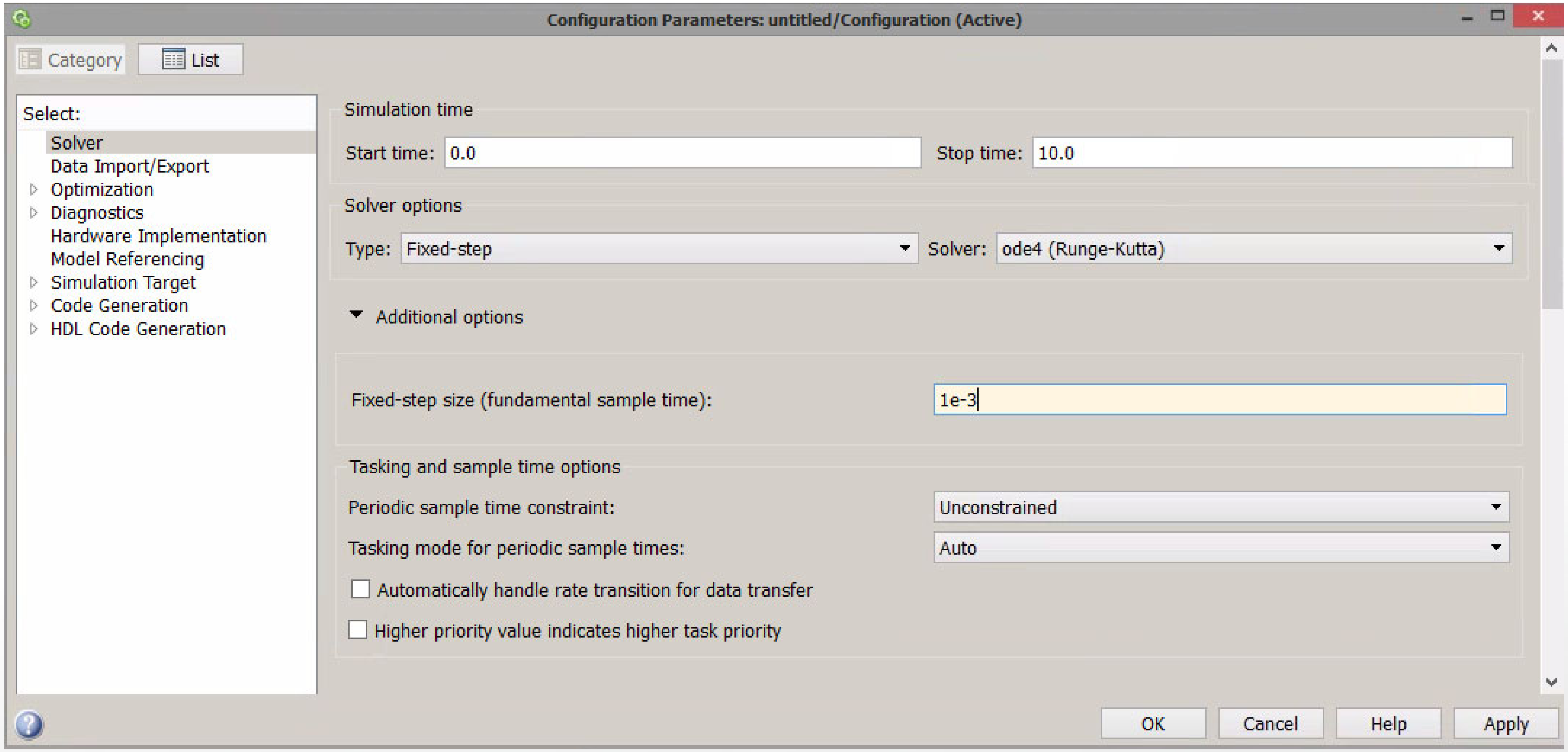
Save 2-D signals as: 3-D array (concatenate along third dimension)

☑ Log fixed-point data as a fi object

Sample time (-1 for inherited): -1

MATLAB / Simulink Example

- And the model configuration parameters are to be set as follows:



Simulation time

Start time: 0.0 Stop time: 10.0

Solver options

Type: Fixed-step Solver: ode4 (Runge-Kutta)

Additional options

Fixed-step size (fundamental sample time): 1e-3

Tasking and sample time options

Periodic sample time constraint: Unconstrained

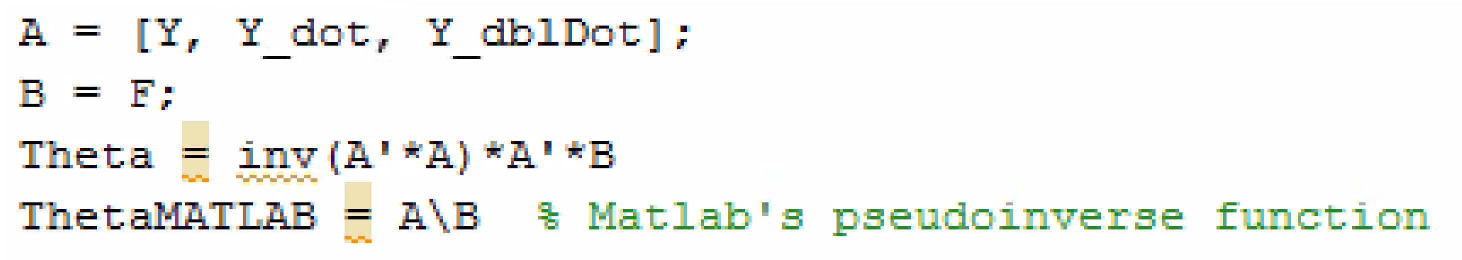
Tasking mode for periodic sample times: Auto

□ Automatically handle rate transition for data transfer

□ Higher priority value indicates higher task priority

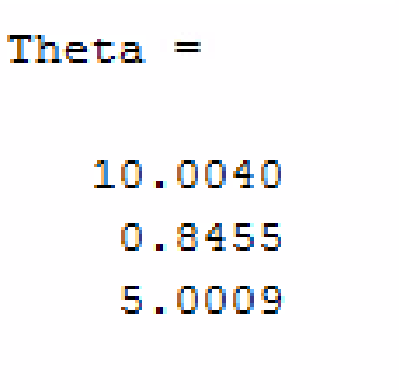
MATLAB / Simulink Example

- In MATLAB, write and run the following codes:

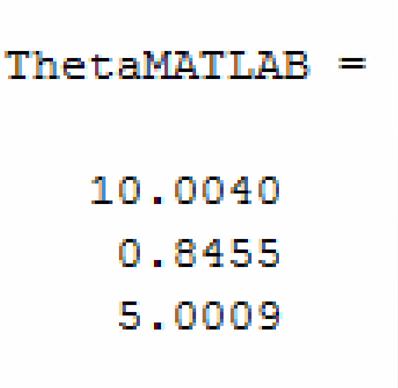


```
A = [Y, Y_dot, Y_dblDot];
B = F;
Theta = inv(A'*A)*A'*B
ThetaMATLAB = A\B  % Matlab's pseudoinverse function
```

- The answer given by MATLAB is:



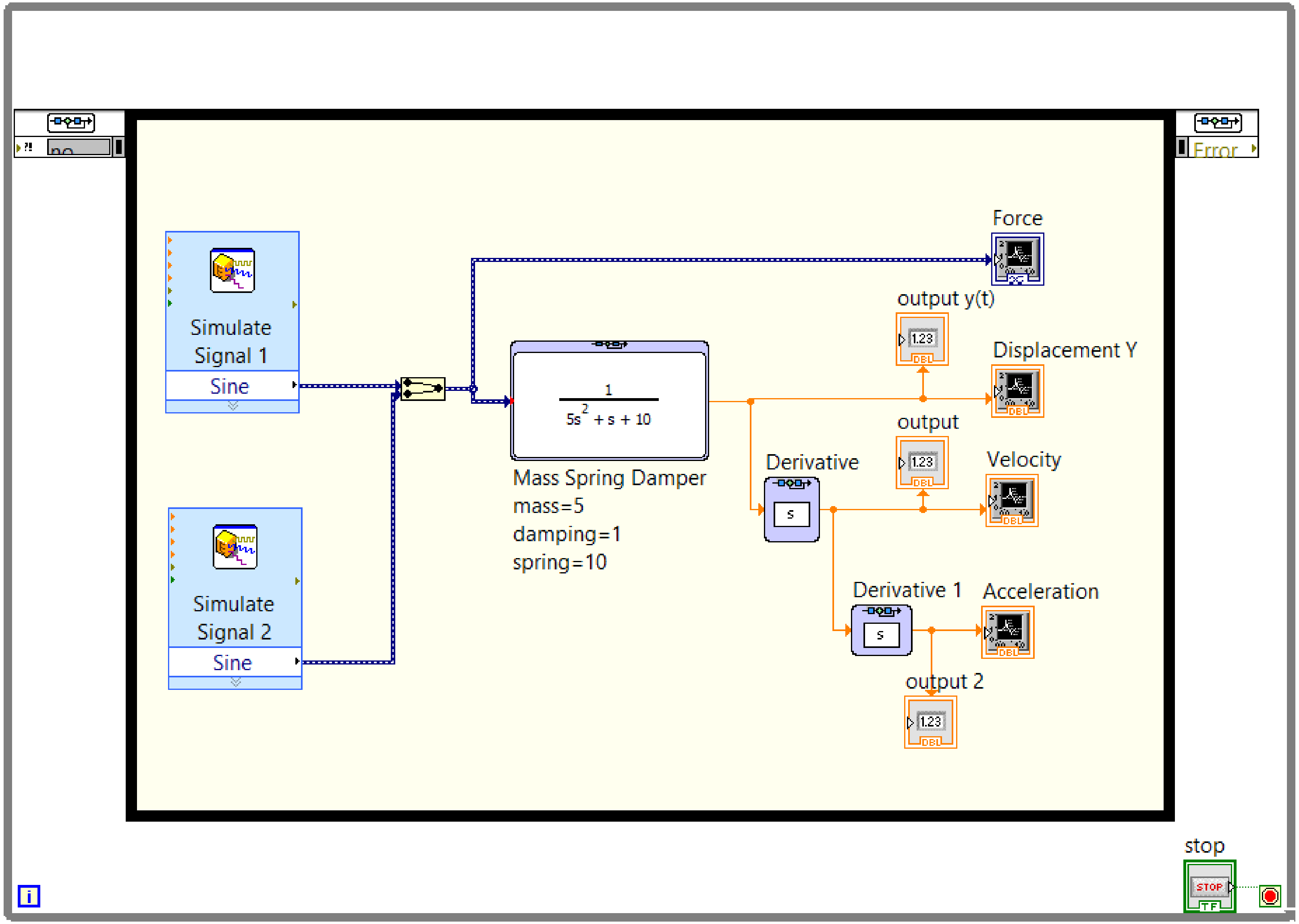
```
Theta =

   10.0040
    0.8455
    5.0009
```



```
ThetaMATLAB =

   10.0040
    0.8455
    5.0009
```

  - Which is close to the real value.

Transfer Functions in LabVIEW



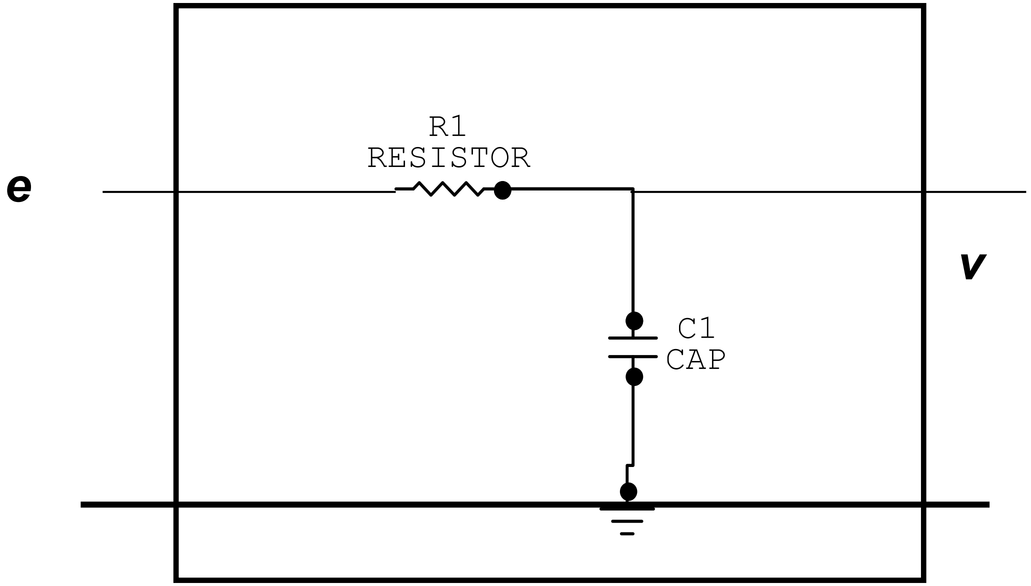
Signal Filtering

- The example shown previously is not practical because of measurement noise.
- If we measure y(t) and then differentiate it, we will amplify the noise.
- Therefore, we should filter both the input and the output before collecting them:

$$\underbrace{\begin{bmatrix} \ddot{y}^f(0) & \dot{y}^f(0) & y^f(0) \\ \ddot{y}^f(T) & \dot{y}^f(T) & y^f(T) \\ \ddot{y}^f(2T) & \dot{y}^f(2T) & y^f(2T) \\ \vdots & \vdots & \vdots \\ \ddot{y}^f(N) & \dot{y}^f(N) & y^f(N) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} m \\ b \\ k \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} F^f(0) \\ F^f(T) \\ F^f(T) \\ \vdots \\ F^f(T) \end{bmatrix}}_{B}$$

- If the filter is the same for both the input and the output, the integrity of the above equation is not affected.

Filter is a
First Order System



$$e = iR + v; \quad i = C\frac{dv}{dt}$$

$$e = C\frac{dv}{dt}R + v$$

$$E(s) = RCsV(s) + V(s)$$

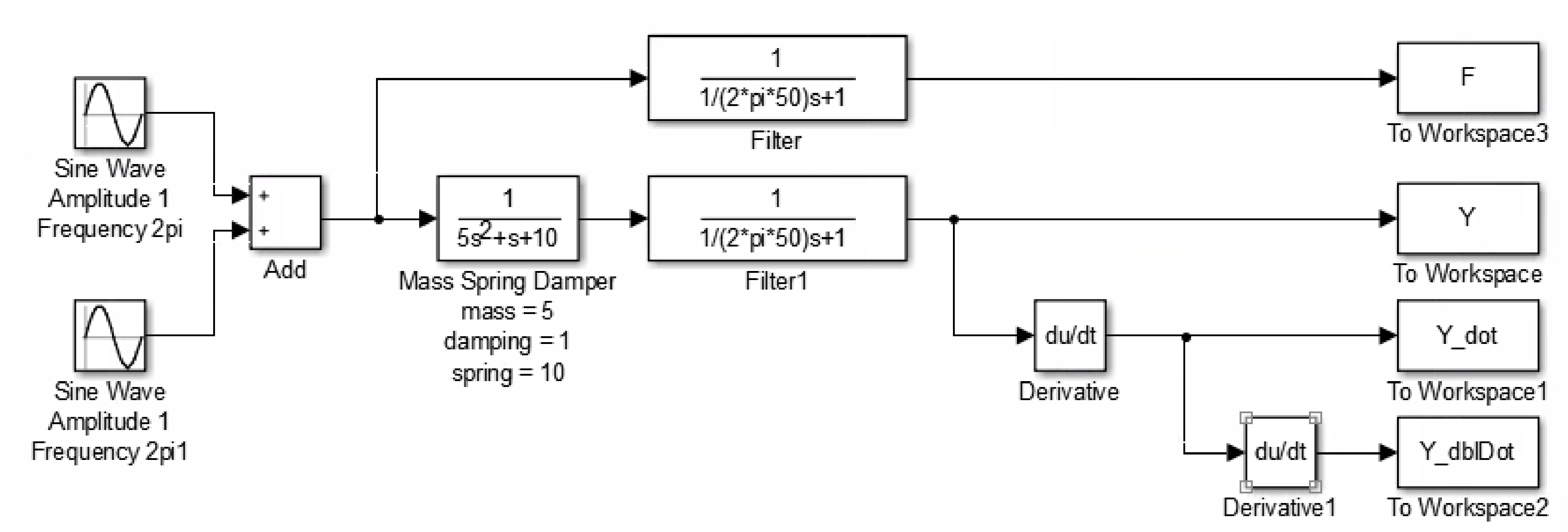
$$\frac{V(s)}{E(s)} = \frac{V(s)}{sRCV(s) + V(s)}$$

$$\frac{V(s)}{E(s)} = \frac{1}{sRC + 1} = \frac{1}{\tau s + 1}$$

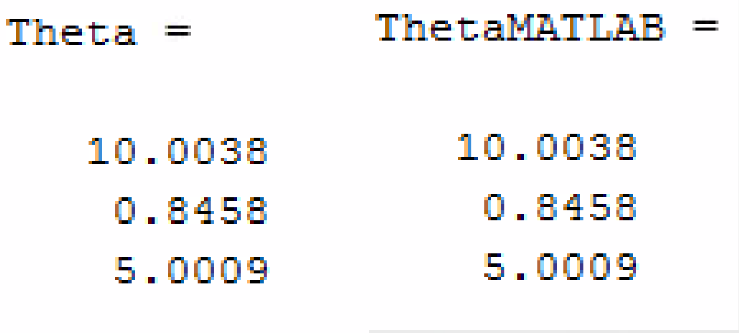
$$\tau = RC = \text{time constant}$$

MATLAB / Simulink Example

- Add in filters in the following Simulink block diagram:



- Run the above model, and then run the MATLAB code for parameter identification:



```
Theta =              ThetaMATLAB =


   10.0038              10.0038
    0.8458               0.8458
    5.0009               5.0009
```

Considerations

- There are a few questions to be answered for the grey-box parameter identification method to work well:
  - If data is acquired digitally, what should the sampling rate be?
  - What is the model structure?
  - How should the excitation signal be?
- First question: Sampling rate
  - Use Nyquist theorem as a guideline for choosing sampling frequency.
    - If signal is band-limited to *fc*, then sampling frequency must be higher than *2fc*.
    - Too high a sampling frequency is not good either:
      - Noise is not filtered
      - Heavy memory use because of much more data

Considerations

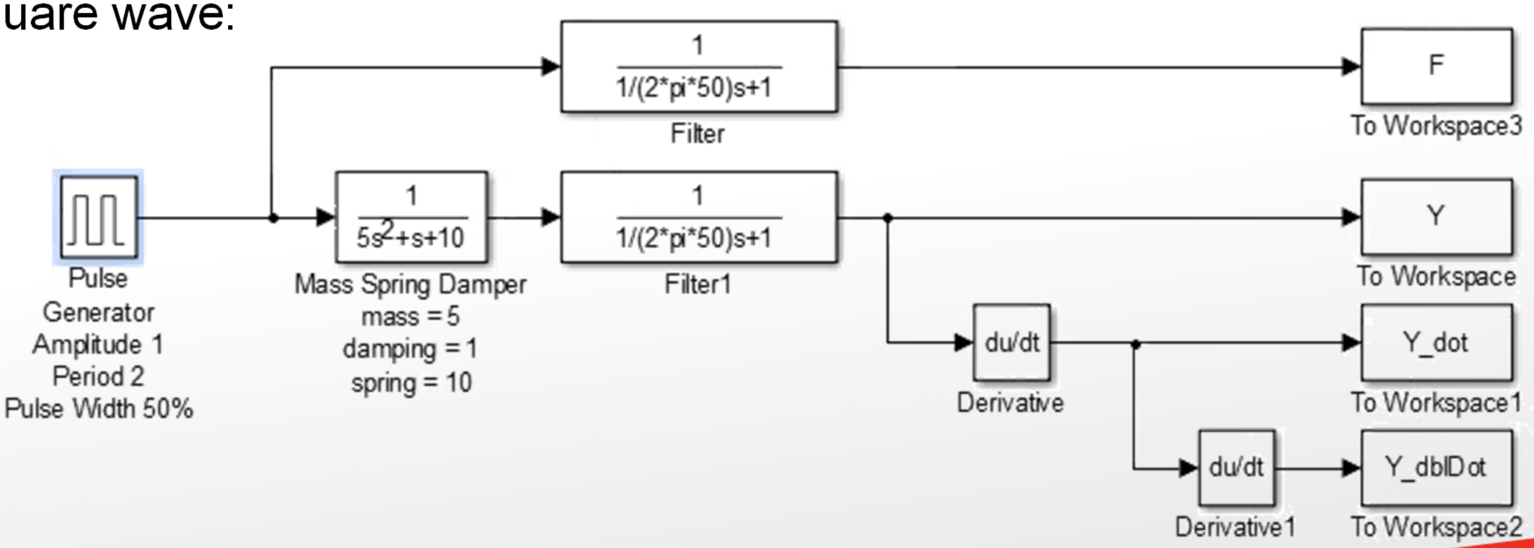
- Second question: Model structure?
  - Not all systems can be identified this way.
  - The physical model (with unknown parameters) should be "Linear-in-Parameters", so that we can separate out the parameters in a vector θ.
    - E.g. $m\ddot{y}(t)+b\dot{y}(t)+ky(t)=F(t)$ → $\begin{bmatrix}\ddot{y}(t) & \dot{y}(t) & y(t)\end{bmatrix}\begin{bmatrix}m\\ b\\ k\end{bmatrix}=F(t)$
  - Some systems which looks nonlinear can actually be written in the Linear-in-Parameters form as well.
    - E.g. $ay(t)+by^2(t)+c\sqrt{y(t)}=u(t)$ → $\begin{bmatrix}y(t) & y^2(t) & \sqrt{y(t)}\end{bmatrix}\begin{bmatrix}a\\ b\\ c\end{bmatrix}=u(t)$
    - E.g.

      $AP^{\beta}e^{u}=Q$ → $\ln A+\beta\ln P+u=\ln Q$ → $\begin{bmatrix}\ln P & 1\end{bmatrix}\begin{bmatrix}\beta\\ u\end{bmatrix}=\ln Q-\ln A$

      assuming A is known.

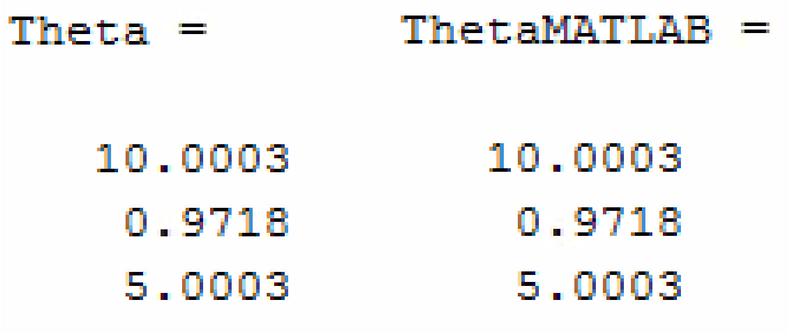
Considerations

- Third question: Excitation signal
  - From the least square inversion $\hat{\theta} = \left(A^T A\right)^{-1} A^T B$, it is clear that the matrix $A^T A$ should be invertible in order to obtain the parameters.
  - But how to make sure that $A^T A$ is invertible?
  - Let's run some experiments to see how the excitation signals should be.
  - Modify the Simulink model just now, by replacing the sine waves with a square wave:



Considerations

- Run the MATLAB code and the result of the parameter identification becomes:

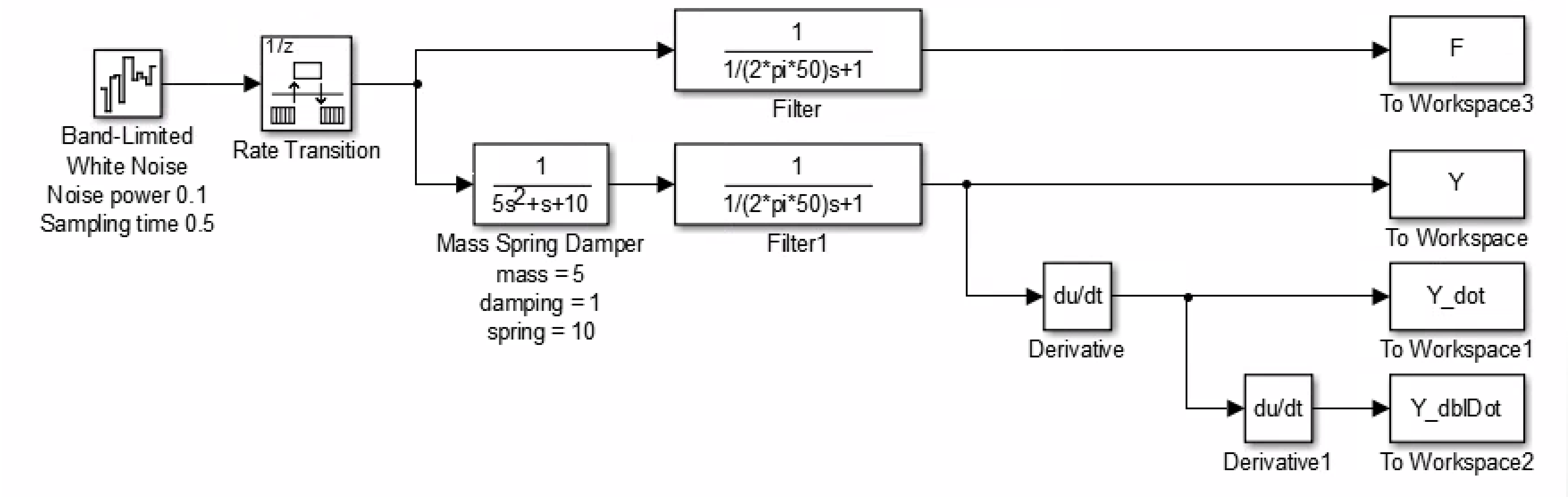


```
Theta =          ThetaMATLAB =


   10.0003          10.0003
    0.9718           0.9718
    5.0003           5.0003
```

- If we compare the squared error of these parameters to those obtained via sine waves, we would see that these new parameters are closer to the real value.

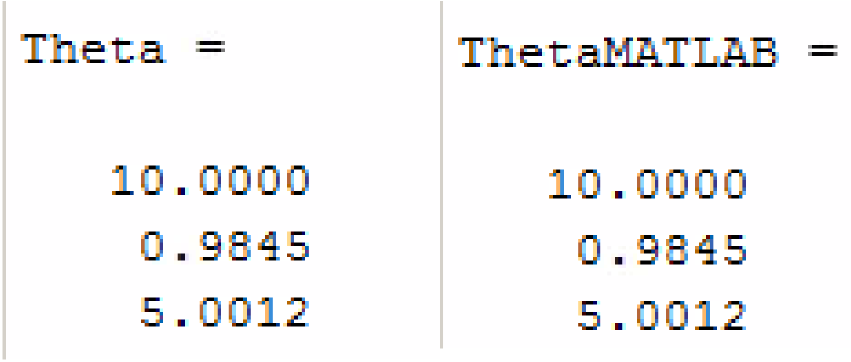
Considerations

- Finally, replace the square wave with the band-limited white noise:



Considerations

- Run the MATLAB code and the result of the parameter identification improves further:



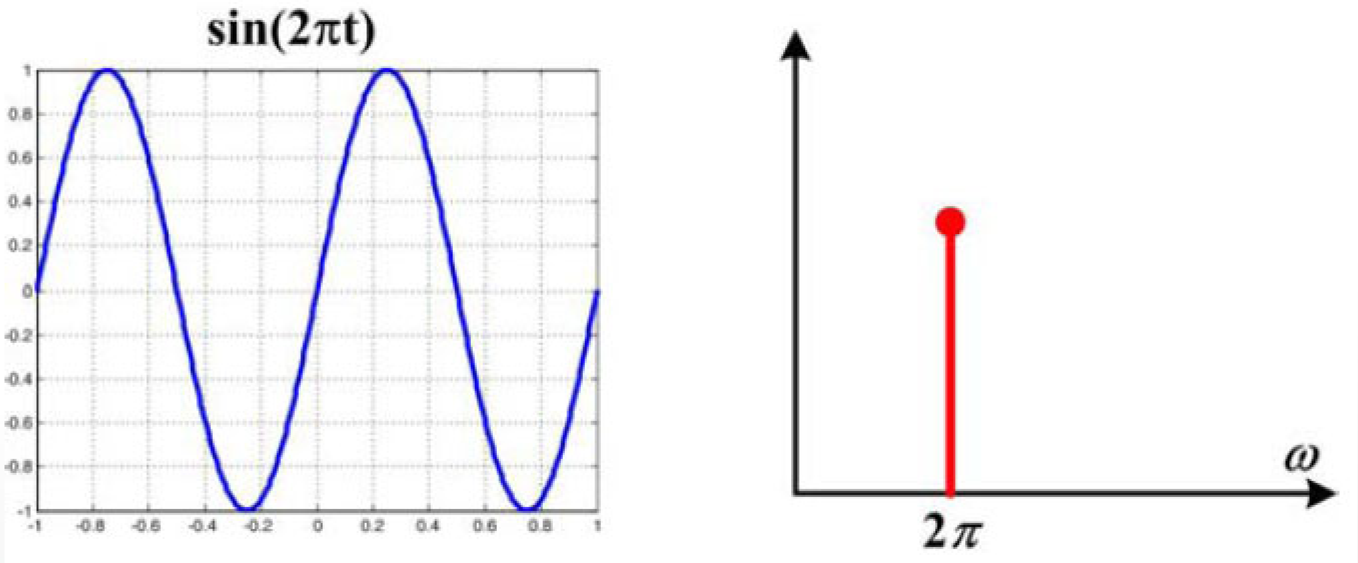
```
Theta =

   10.0000
    0.9845
    5.0012
```

```
ThetaMATLAB =

   10.0000
    0.9845
    5.0012
```

- In this example, the systems is fairly simple, and there are only three parameters to identify. The sine waves, square waves and white noise, and in fact a step input, would give good results.
- In general, the more complex (“rich”) the input signal is, the better the quality of identification becomes.
- Continue...

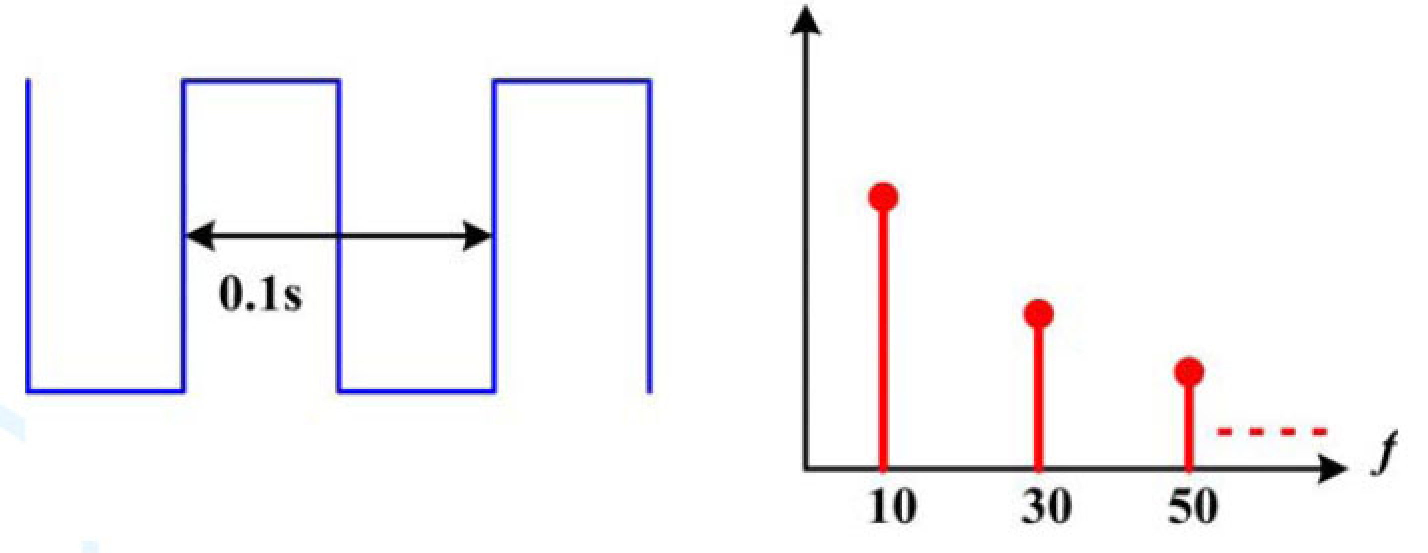
Considerations

- The “richness” of the signal can be quantified by its spectrum:
  - Spectrum is the frequency domain representation of a time-domain signal.
  - Fourier transform: $f(t) \Rightarrow F(\omega)$
  - A sine wave $A\sin(\omega_0 t)$ contains only one frequency:



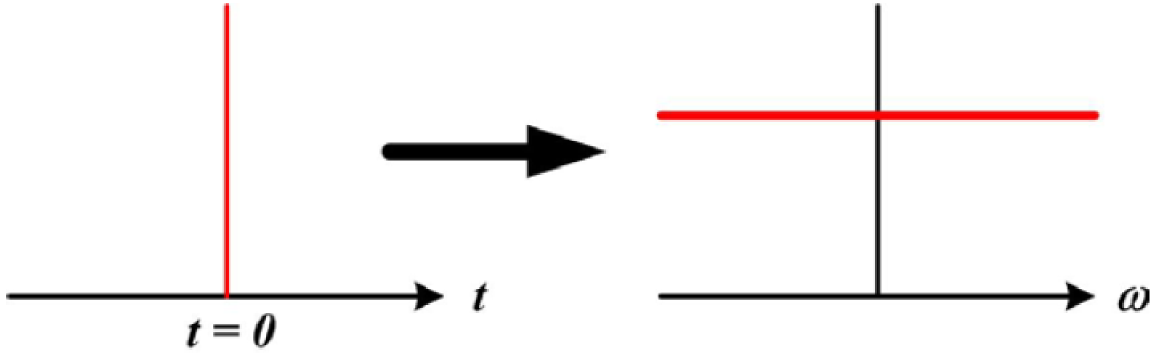
Considerations

- A square wave of 10Hz has the frequency contents at 10Hz, 30Hz, 50Hz etc. With diminishing amplitude:

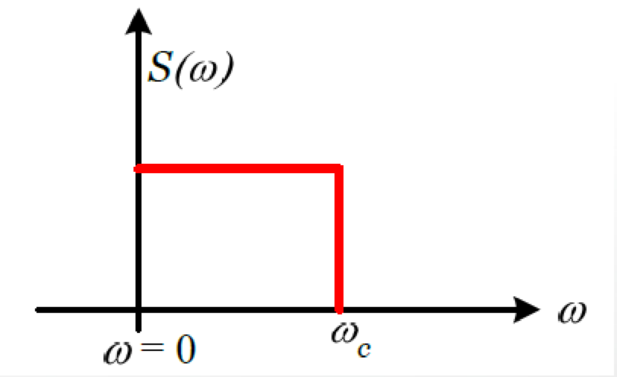


Considerations

- A random signal has a flat frequency spectrum:



- Computer generated random signal (also called band-limited white noise):



Summary: To capture the system's frequency domain property up to a frequency ω_0, then the magnitude of the input spectrum must not be too small for all frequencies up to ω_0

- Often used for system identification.

Example: DC Motor Speed

- In week 5, we learnt that the relationship between input voltage and angular speed of a DC motor is:

$$\frac{\Omega}{V}=\frac{k_t}{(Js+b)(Ls+R)+k_tk_e}$$
$$=\frac{k_t}{JLs^2+(JR+bL)s+(bR+k_tk_e)}$$
$$=\frac{1}{\frac{JL}{k_t}s^2+\frac{(JR+bL)}{k_t}s+\frac{(bR+k_tk_e)}{k_t}}$$

(Laplace transform will be introduced shortly. At this stage just see “s” as a differential operator.)

- Therefore:

$$\frac{JL}{k_t}\ddot{\omega}+\frac{(JR+bL)}{k_t}\dot{\omega}+\frac{(bR+k_tk_e)}{k_t}\omega=V$$

- This has the same form as the mass-spring-damper system, but instead of identifying each physical parameter, we may have to be content with identifying “combined” parameters.

Example: DC Motor Speed

- We excite the DC motor with a varying voltage V, and then collect the values of V, speed and its derivatives, and put all the collected data in the following form:

$$\underbrace{\begin{bmatrix} \ddot{\omega}(0) & \dot{\omega}(0) & \omega(0) \\ \ddot{\omega}(T) & \dot{\omega}(T) & \omega(T) \\ \ddot{\omega}(2T) & \dot{\omega}(2T) & \omega(2T) \\ \vdots & \vdots & \vdots \\ \ddot{\omega}(N) & \dot{\omega}(N) & \omega(N) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \frac{JL}{k_t} \\ \frac{(JR+bL)}{k_t} \\ \frac{(bR+k_t k_e)}{k_t} \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} V(0) \\ V(T) \\ V(2T) \\ \vdots \\ V(N) \end{bmatrix}}_{B}$$

Transpose Matrix

$$A= \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \qquad A^T= \begin{matrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{matrix}$$

- By using the least squares method, $\hat{\theta} = \left(A^T A\right)^{-1} A^T B$, we can estimate the parameters θ, which consists of the combination of the real physical parameters.

Contents

- Introduction
- Identification of Continuous-Time Models in Time Domain
  - Grey Box
  - Black Box
- Identification of Discrete-Time Models
  - Grey Box
  - Black Box
- Identification of Continuous-Time Models in Frequency Domain

Black-Box Modelling

- So far, we have been dealing with grey-box modelling, i.e. the form of the model is known and we needed only to estimate the parameters.
- What if we do not even have the form of the model to start with?
  - We need to assume a model structure!
  - In general, any continuous-time dynamic system can be described by the ordinary differential equation (ODE):

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u$$

  - A few terminologies:
    - Order of the system: The highest derivative of the output y, i.e. n
    - Relative order of the system: n – m
  - But what should *n* and *m* be?

Black-Box Modelling

- Systematic way:
  - Estimate *n* based on step response. If step response is monotonously increasing to constant level, then *n* is probably 1. If step response resembles standard 2nd order system response, then *n* is probably 2. Otherwise *n* could be higher.
  - Choose a range of *n* and *m* that includes the estimated *n*.
  - Run the experiment for all combinations of *n* and *m*.
  - Calculate cost function V and choose the set {*n*, *m*} which gives the least value.
- Additionally, we can test the above combinations, and the redundant parameters will have values close to zero.
- All other questions for grey box modelling (e.g. Sample rate, excitation signals etc.) are still valid for black box modelling.

Black-Box Modelling

- Example: Consider the same mass-spring-damper system again.
- However, assume that we do not know the form of the model.
- We will therefore use the general form of the ODE:

$$a_n \frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m u}{dt^m} + b_{m-1}\frac{d^{m-1}u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u$$

- We can first normalize all the parameters with a_0 and obtain the following:

$$-\tilde{a}_n \frac{d^n y}{dt^n} - \tilde{a}_{n-1}\frac{d^{n-1}y}{dt^{n-1}} - \cdots - \tilde{a}_1 \frac{dy}{dt} + \tilde{b}_m \frac{d^m u}{dt^m} + \tilde{b}_{m-1}\frac{d^{m-1}u}{dt^{m-1}} + \cdots + \tilde{b}_1 \frac{du}{dt} + \tilde{b}_0 u = y$$

- And we will try different values for *n* = 1 to 3 and *m* = 0 to 1.
- That means, we should collect

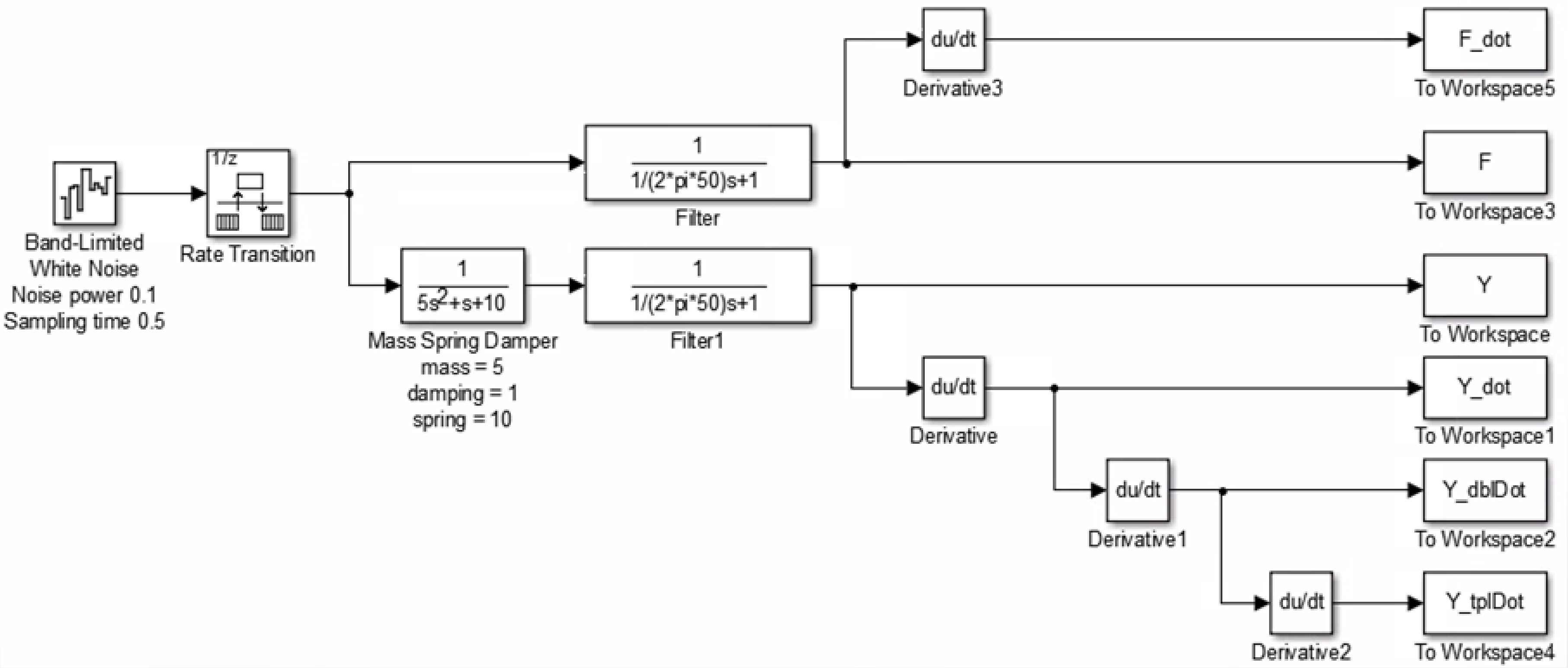
$$y(t), \dot{y}(t), \ddot{y}(t), \dddot{y}(t), u(t), \dot{u}(t)$$

  - Or preferably the filtered signals:

$$y^f(t), \dot{y}^f(t), \ddot{y}^f(t), \dddot{y}^f(t), u^f(t), \dot{u}^f(t)$$

Black-Box Modelling

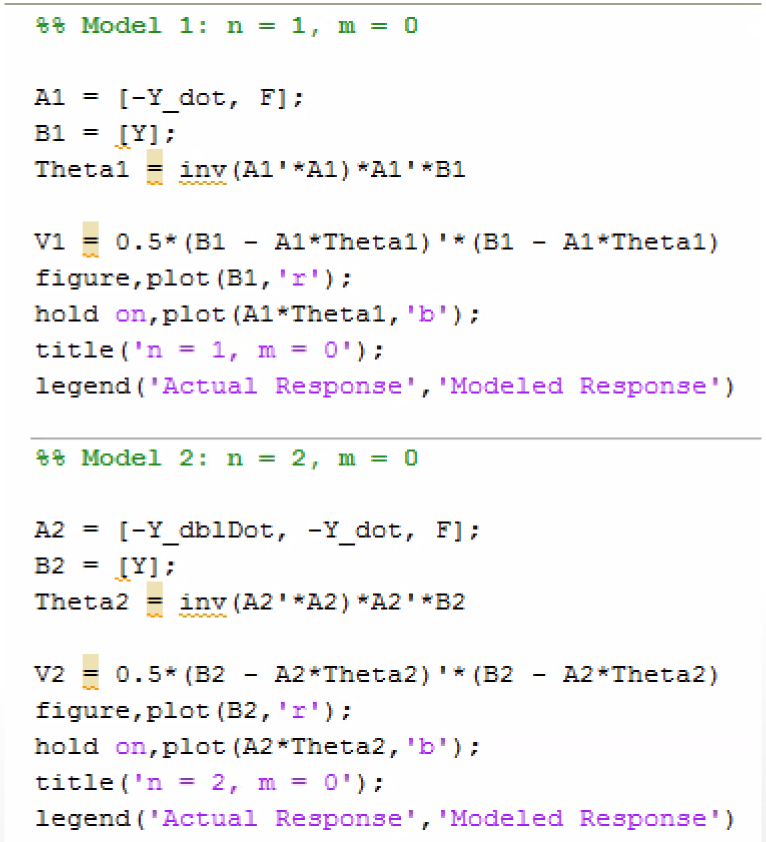
- Modify the previous Simulink model as follows:



- Run the model.

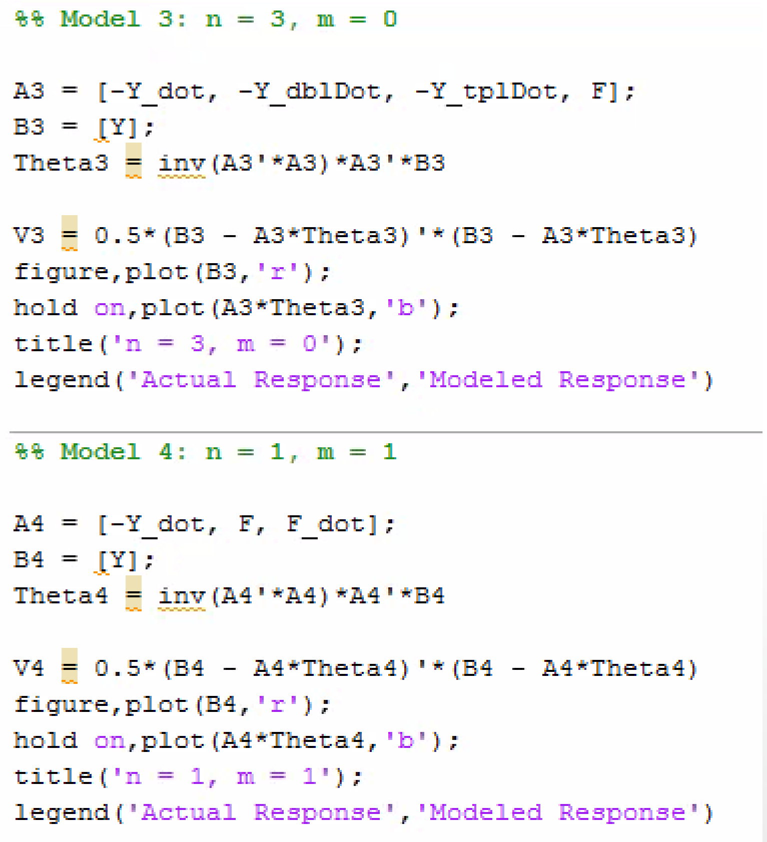
Black-Box Modelling

- Next, write a MATLAB code as follows:



```
%% Model 1: n = 1, m = 0

A1 = [-Y_dot, F];
B1 = [Y];
Theta1 = inv(A1'*A1)*A1'*B1

V1 = 0.5*(B1 - A1*Theta1)'*(B1 - A1*Theta1)
figure,plot(B1,'r');
hold on,plot(A1*Theta1,'b');
title('n = 1, m = 0');
legend('Actual Response','Modeled Response')
```

```
%% Model 2: n = 2, m = 0

A2 = [-Y_dblDot, -Y_dot, F];
B2 = [Y];
Theta2 = inv(A2'*A2)*A2'*B2

V2 = 0.5*(B2 - A2*Theta2)'*(B2 - A2*Theta2)
figure,plot(B2,'r');
hold on,plot(A2*Theta2,'b');
title('n = 2, m = 0');
legend('Actual Response','Modeled Response')
```

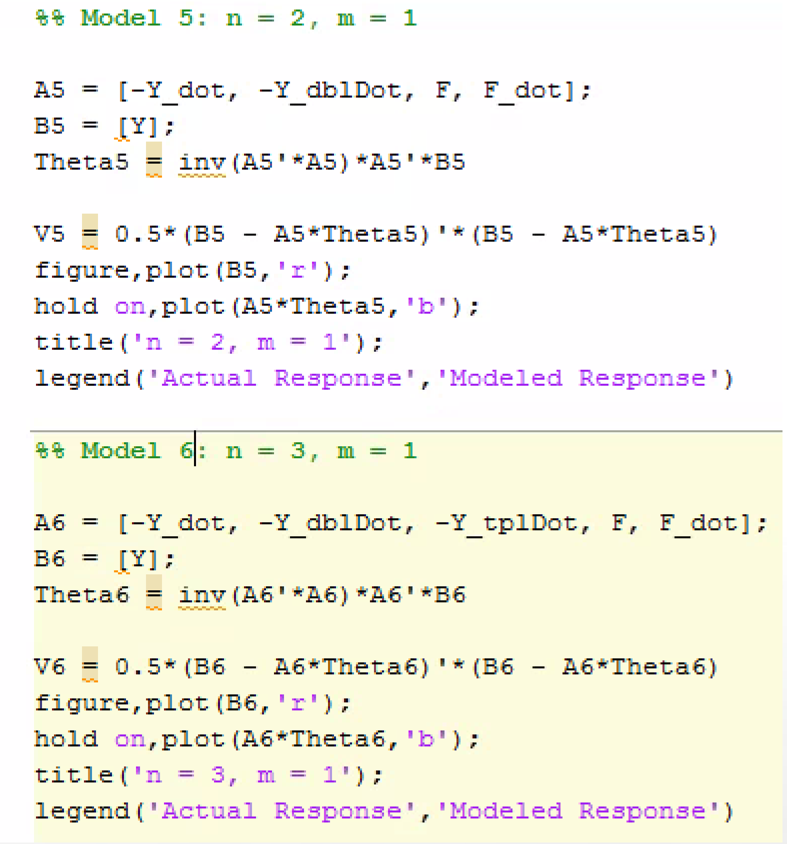


```
%% Model 3: n = 3, m = 0

A3 = [-Y_dot, -Y_dblDot, -Y_tplDot, F];
B3 = [Y];
Theta3 = inv(A3'*A3)*A3'*B3

V3 = 0.5*(B3 - A3*Theta3)'*(B3 - A3*Theta3)
figure,plot(B3,'r');
hold on,plot(A3*Theta3,'b');
title('n = 3, m = 0');
legend('Actual Response','Modeled Response')
```

```
%% Model 4: n = 1, m = 1

A4 = [-Y_dot, F, F_dot];
B4 = [Y];
Theta4 = inv(A4'*A4)*A4'*B4

V4 = 0.5*(B4 - A4*Theta4)'*(B4 - A4*Theta4)
figure,plot(B4,'r');
hold on,plot(A4*Theta4,'b');
title('n = 1, m = 1');
legend('Actual Response','Modeled Response')
```

Black-Box Modelling

- And:



```
%% Model 5: n = 2, m = 1

A5 = [-Y_dot, -Y_dblDot, F, F_dot];
B5 = [Y];
Theta5 = inv(A5'*A5)*A5'*B5

V5 = 0.5*(B5 - A5*Theta5)'*(B5 - A5*Theta5)
figure,plot(B5,'r');
hold on,plot(A5*Theta5,'b');
title('n = 2, m = 1');
legend('Actual Response','Modeled Response')
```

```
%% Model 6: n = 3, m = 1

A6 = [-Y_dot, -Y_dblDot, -Y_tplDot, F, F_dot];
B6 = [Y];
Theta6 = inv(A6'*A6)*A6'*B6

V6 = 0.5*(B6 - A6*Theta6)'*(B6 - A6*Theta6)
figure,plot(B6,'r');
hold on,plot(A6*Theta6,'b');
title('n = 3, m = 1');
legend('Actual Response','Modeled Response')
```

Black-Box Modelling

- Run the algorithm and we will obtain:
  - V1 = 23.4447
  - V2 = 0.0065
  - V3 = 0.0016
  - V4 = 23.4352
  - V5 = 7.9848e-5
  - V6 = 2.9429e-6
- It seems as though models 3, 5 and 6 are the best. However, if we look at the parameters, we will find that there are parameters which are close to zero.
- Thus probably model 2 is considered good too, which is identified as:

$$-0.5\frac{d^2y}{dt^2}-0.0982\frac{dy}{dt}+0.0999u=y$$

$$5\frac{d^2y}{dt^2}+0.982\frac{dy}{dt}+10y=0.999u$$

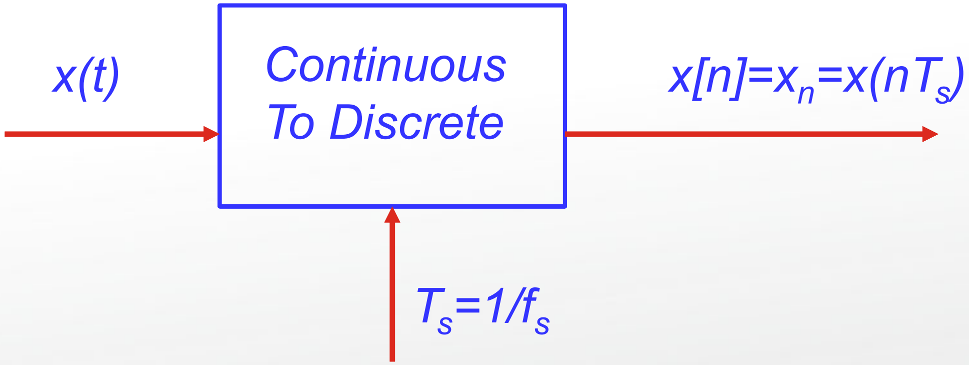
- And is similar to the true system.

Contents

- Introduction
- Identification of Continuous-Time Models in Time Domain
- Identification of Continuous-Time Models in Frequency Domain
- Identification of Discrete-Time Models
    - Grey Box
    - Black Box
- Identification of Continuous-Time Models in Frequency Domain

Discrete-Time Model

- We already learnt that most data acquisition system can only acquire data at discrete values of time.
  - The system which is originally continuous “becomes” a discrete-time system from the viewpoint of the computer or microprocessor.
  - Therefore, it is worthwhile to know how to identify discrete-time models.



Difference Equation Examples

$$y_{k+1} = 2y_{k-2} + 3y_{k-1} - 4y_k;$$

where

$$k = \{2,3,4,\ldots n\}$$

$$y_0 = 0$$

$$y_1 = 3$$

$$y_2 = 1$$

Generate following 8 function values, from $y_3 - y_{10}$

- Use table representation of the function

- Use graph representation

Difference Equation

$$y_k = 2y_{k-3} + 3y_{k-2} - 4y_{k-1};$$

where

$$k = \{2,3,4,\ldots n\}$$

$$y_0 = 0$$

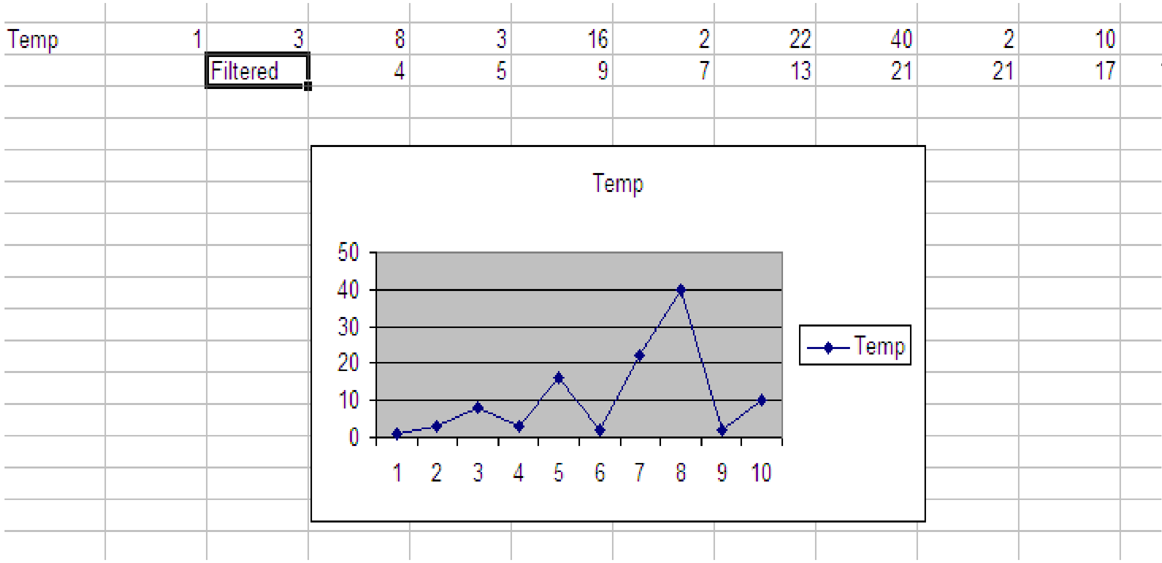
$$y_1 = 3$$

$$y_2 = 1$$

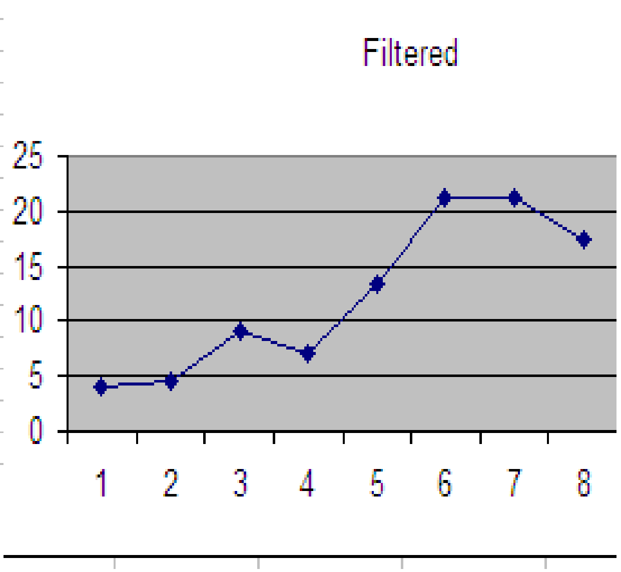
0	1	2	3	4	5	6	7	8	9	10
y_o	y_1	y_2								
0	3	1								

FIR Filtering - Low Pass Filtering of a Discrete Quantity

$$y_k = \frac{x_{k-2} + x_{k-1} + x_k}{3}$$



Temp	1	3	8	3	16	2	22	40	2	10
		Filtered	4	5	9	7	13	21	21	17



Finite Impulse Response (FIR)

Discrete-Time Model

- Example: Mass-Spring-Damper system:
- Continuous-time model: $m\ddot{y}+b\dot{y}+ky=u$
- The derivatives can be approximated as:

$$\dot{y}\approx\frac{y(k)-y(k-1)}{T}$$

$$\ddot{y}\approx\frac{\dot{y}(k)-\dot{y}(k-1)}{T}\approx\frac{\frac{y(k)-y(k-1)}{T}-\frac{y(k-1)-y(k-2)}{T}}{T}=\frac{y(k)-2y(k-1)+y(k-2)}{T^2}$$

Discrete-Time Model

- Therefore, the continuous-time model can be approximated as the following discrete-time model:

$$m\frac{y(k)-2y(k-1)+y(k-2)}{T^2}+b\frac{y(k)-y(k-1)}{T}+ky(k)=u(k)$$

$$\left(\frac{m}{T^2}+\frac{b}{T}+k\right)y(k)-\left(\frac{2m}{T^2}+\frac{b}{T}\right)y(k-1)+\frac{m}{T^2}y(k-2)=u(k)$$

$$y(k)=\underbrace{\left(\frac{\frac{2m}{T^2}+\frac{b}{T}}{\frac{m}{T^2}+\frac{b}{T}+k}\right)}_{a_1}y(k-1)+\underbrace{\left(\frac{-\frac{m}{T^2}}{\frac{m}{T^2}+\frac{b}{T}+k}\right)}_{a_2}y(k-2)+\underbrace{\left(\frac{1}{\frac{m}{T^2}+\frac{b}{T}+k}\right)}_{b_0}u(k)$$

Identification of Discrete-Time Model

- Because the discrete-time model is already in the linear-in-parameter form, identification of discrete-time system is rather straightforward.
- Excite the system with varying input u.
- Collect the input and output at each sampling time, and build the following equation:

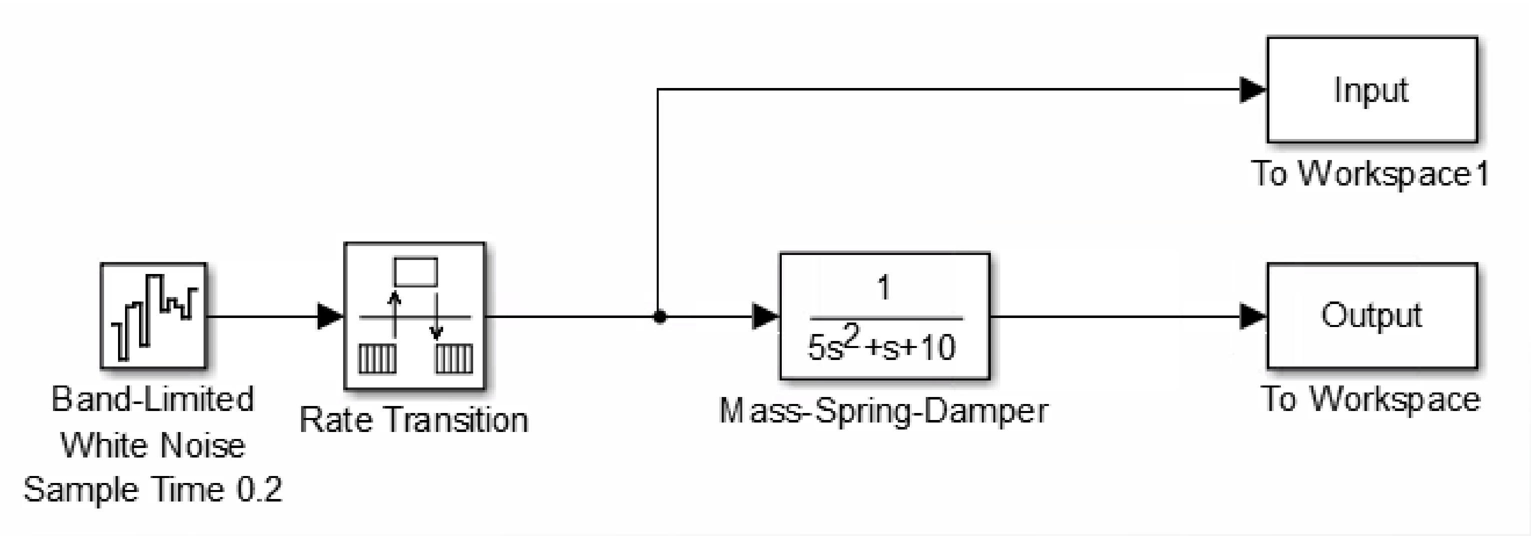
$$\underbrace{\begin{bmatrix} y(-1) & y(-2) & \cdots & u(0) & u(-1) & \cdots \\ y(0) & y(-1) & \cdots & u(1) & u(0) & \cdots \\ y(1) & y(0) & \cdots & u(2) & u(1) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ y(N-1) & y(N-2) & \cdots & u(N) & u(N-1) & \cdots \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ b_0 \\ b_1 \\ \vdots \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}}_{B}$$

- The parameters are then calculated using the Least Squares method:

$$\hat{\theta} = \left(A^T A\right)^{-1} A^T B$$

MATLAB Simulink Example

- We will use the mass-spring-damper system again.
- Create the following model in Simulink:



- Run the model and the input-output data will be saved to Workspace.
- In real experiments, you can use LabVIEW to acquire the data.

MATLAB Simulink Example – Grey Box

- The discretized model of the mass-spring-damper system was:

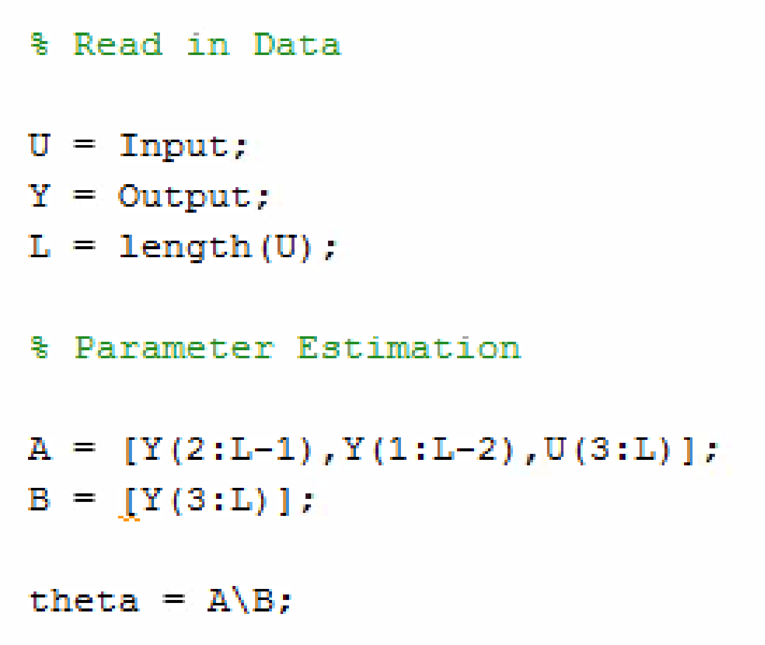
$$y(k) = a_1 y(k-1) + a_2 y(k-2) + b_0 u(k)$$

- Therefore, we can build the following equations to estimate the parameters:

$$\begin{bmatrix} y(1) & y(0) & u(2) \\ y(2) & y(1) & u(3) \\ y(3) & y(2) & u(4) \\ \vdots & \vdots & \vdots \\ y(N-1) & y(N-2) & u(N) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_0 \end{bmatrix} = \begin{bmatrix} y(2) \\ y(3) \\ y(4) \\ \vdots \\ y(N) \end{bmatrix}$$

MATLAB Simulink Example – Grey Box

- This can be done in MATLAB as follows:



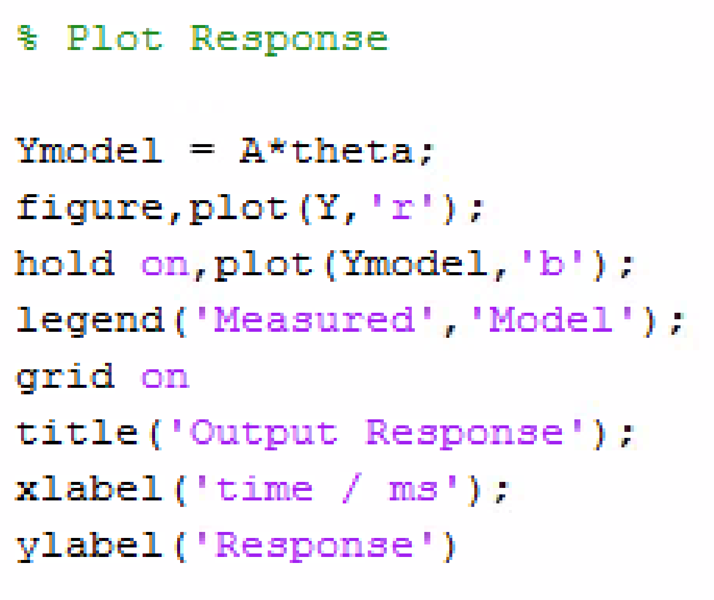
```
% Read in Data

U = Input;
Y = Output;
L = length(U);

% Parameter Estimation

A = [Y(2:L-1),Y(1:L-2),U(3:L)];
B = [Y(3:L)];

theta = A\B;
```

- The parameters are identified as: and are close to the real value.

$$a_1 = 1.9998$$
$$a_2 = -0.9998$$
$$b_0 = 1.984 \times 10^{-7}$$

MATLAB Simulink Example – Grey Box

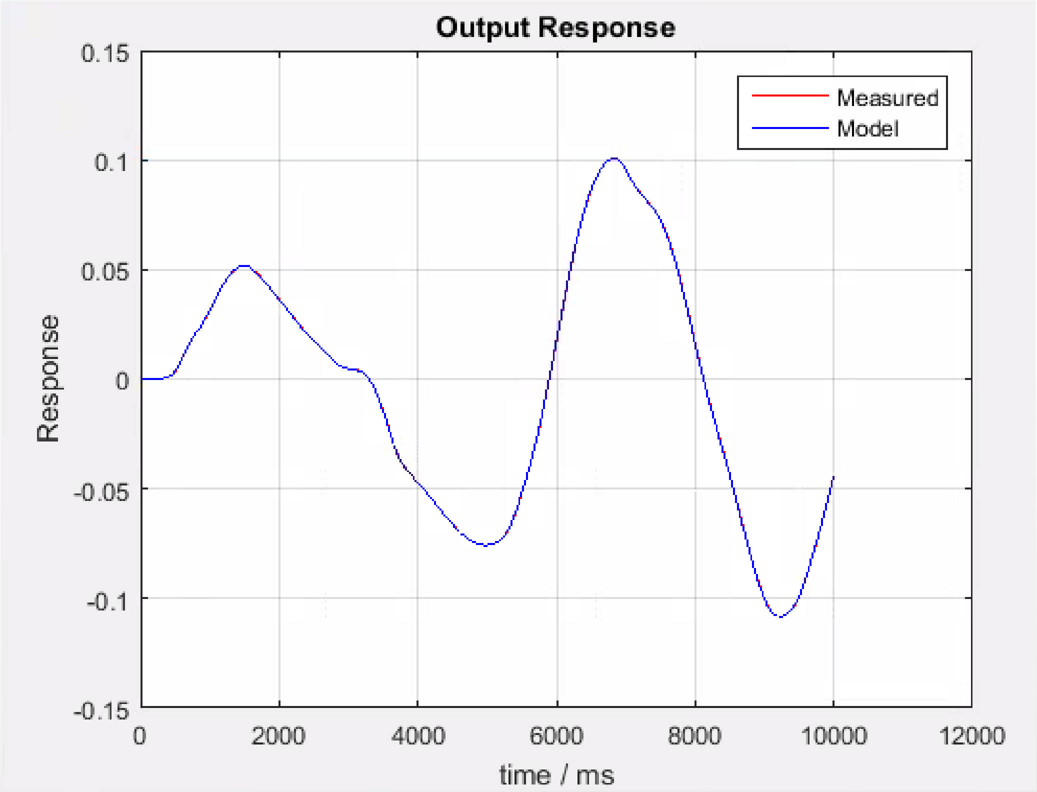
- In real experiments with unknown systems, we will not be able to know the actual parameters for verification purpose.
- To know how good our model is, we can plot the real response vs. The response obtained from the model.
- MATLAB code:



```
% Plot Response

Ymodel = A*theta;
figure,plot(Y,'r');
hold on,plot(Ymodel,'b');
legend('Measured','Model');
grid on
title('Output Response');
xlabel('time / ms');
ylabel('Response')
```

MATLAB Simulink Example – Grey Box

- As can be seen, the modelled output resembles the real output well.
- Therefore, the model is accurate.



Contents

- Introduction
- Identification of Continuous-Time Models in Time Domain
- Identification of Continuous-Time Models in Frequency Domain
- Identification of Discrete-Time Models
  - Grey Box
  - Black Box
- Identification of Continuous-Time Models in Frequency Domain

Linear Differential Equation Model

- Recall that continuous-time dynamic system can in general be described by the ordinary differential equation (ODE).

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u$$

- Similarly, discrete-time systems are described by the linear difference equation model.

$$\begin{aligned} y(k) = {} & a_1 y(k-1) + a_2 y(k-2) + \cdots + a_n y(k-n) \\ & + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_m u(k-m) \end{aligned}$$

$$y(k) = \sum_{i=1}^{n} a_i y(k-1) + \sum_{j=0}^{m} b_j u(k-j)$$

Black Box

- If we do not know the form of the model, then we will have to identify the "Black Box" between the input and output.

$$\begin{aligned} y(k) &= a_1 y(k-1) + a_2 y(k-2) + \cdots + a_n y(k-n) \\ &\quad + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_m u(k-m) \end{aligned}$$

$$\underbrace{\begin{bmatrix} y(-1) & y(-2) & \cdots & u(0) & u(-1) & \cdots \\ y(0) & y(-1) & \cdots & u(1) & u(0) & \cdots \\ y(1) & y(0) & \cdots & u(2) & u(1) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ y(N-1) & y(N-2) & \cdots & u(N) & u(N-1) & \cdots \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ b_0 \\ b_1 \\ \vdots \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}}_{B}$$

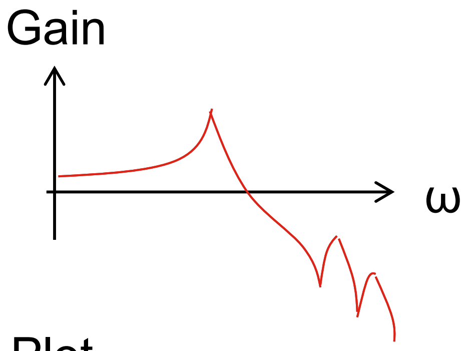
- We will have the same questions to answer just like for the continuous-time case, one of which is the order of n and m.
  - Similar to continuous-time case, we can test different orders of the system, and then pick the best model amongst all.

Contents

- Introduction
- Identification of Continuous-Time Models in Time Domain
- Identification of Discrete-Time Models
  - Grey Box
  - Black Box
- Identification of Continuous-Time Models in Frequency Domain

Frequency Response Model

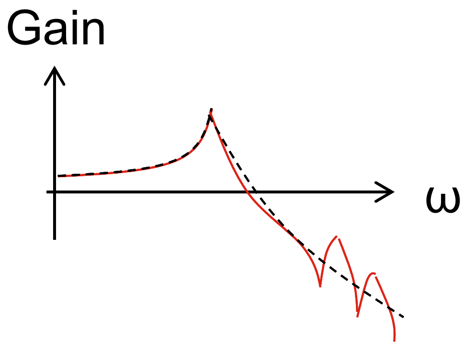
- In Lecture 9, you have encountered the frequency response model of a piezo actuator.
- Graphically, it looks as follows:



  - This is called a Bode Plot.

Frequency Response Model

- We can fit a “transfer function” to the measured Bode Plot, to get an approximation over a range of frequencies:



- Red (solid): Measured
- Black (dashed): Transfer function model

$$G(s) = \frac{\omega^2}{s^2 + 2\xi\omega s + \omega^2}$$

Frequency Response Model

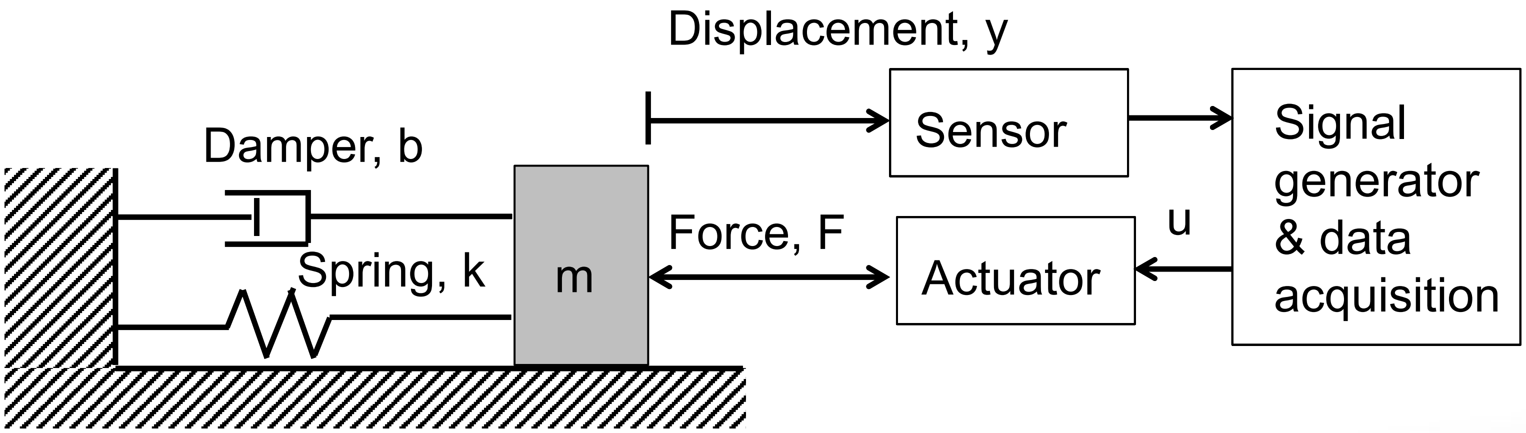
- Questions:
- 1) What does the Bode plot represent? How to interpret it?
- 2) How can we obtain the Bode plot of a system?
- 3) How do we fit a transfer function model?

Q1: What is Bode Plot

- If a sine wave of frequency ω is applied at the input of a linear system, then the output is also a sine wave of frequency ω.
- However:
  - The amplitudes of the input sine wave and the output sine wave are probably different.
    - The ratio between the amplitudes of the output and input is called the “Gain” of the system.
  - The output sinusoid may be phase-shifted from the input sinusoid.
  - Both the gain and phase-shift are frequency dependent.

Q1: What is Bode Plot

- E.g. Mass-Spring-Damper System:



Displacement, y

Damper, b

Spring, k

m

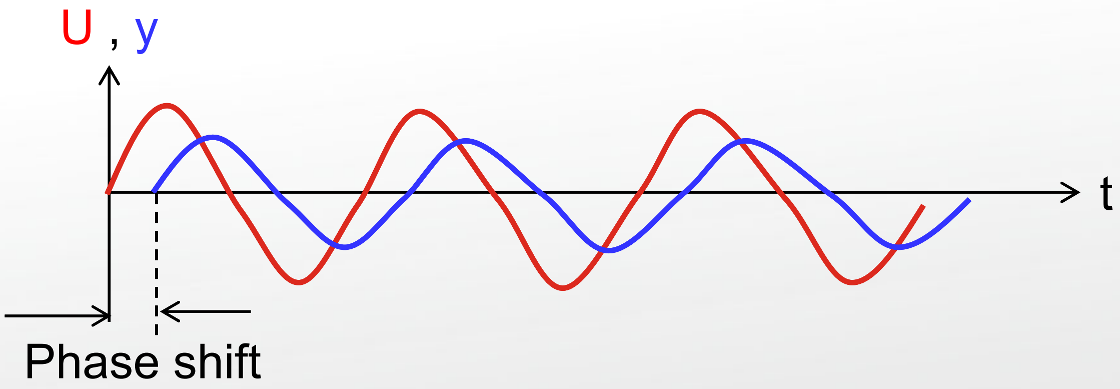
Force, F

Sensor

Actuator

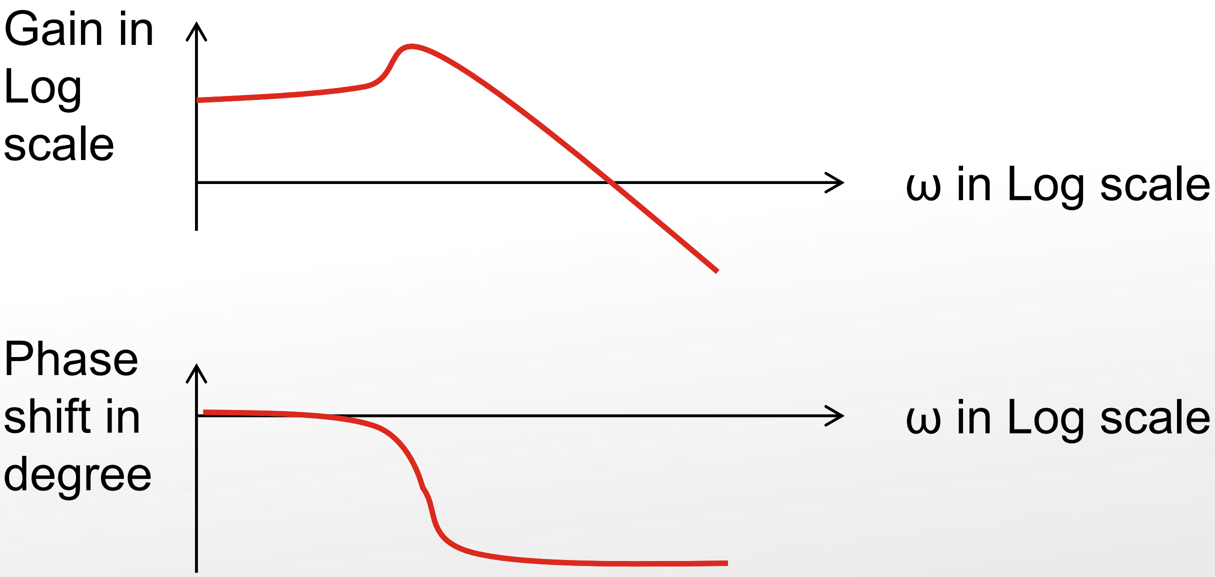
u

Signal generator & data acquisition



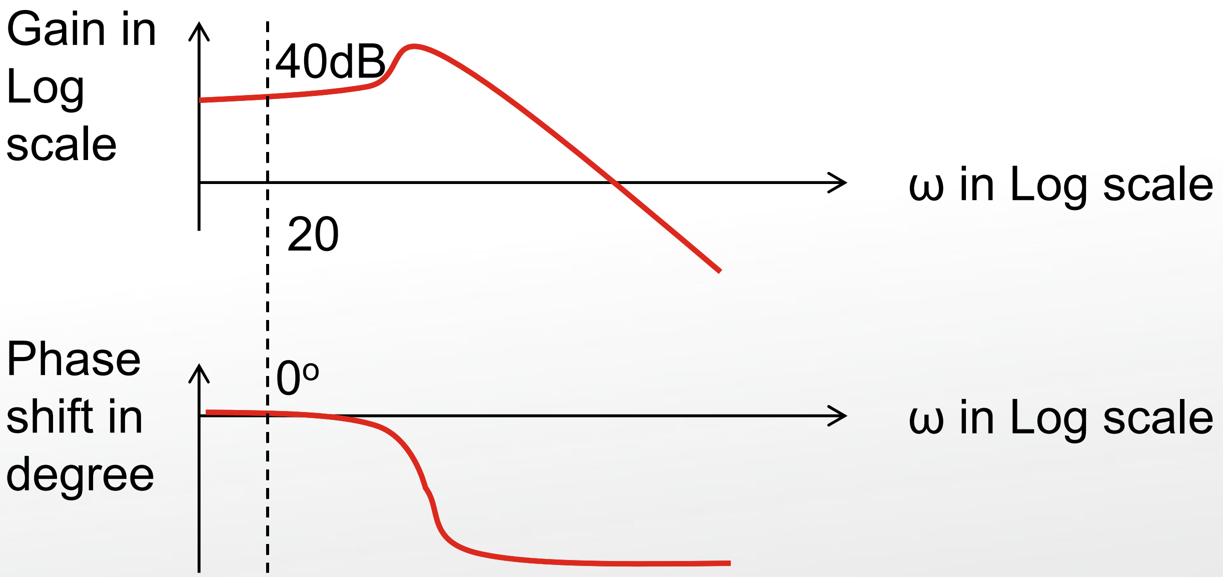
Q1: What is Bode Plot

- Now, if we excite the system with a sinusoidal input of a range of frequencies, and we measure the gain and phase-shift at each of the frequency, we can characterize the system as a function of frequency.
  - This is known as the "Frequency Response".
  - This can be drawn in a "Bode Plot":



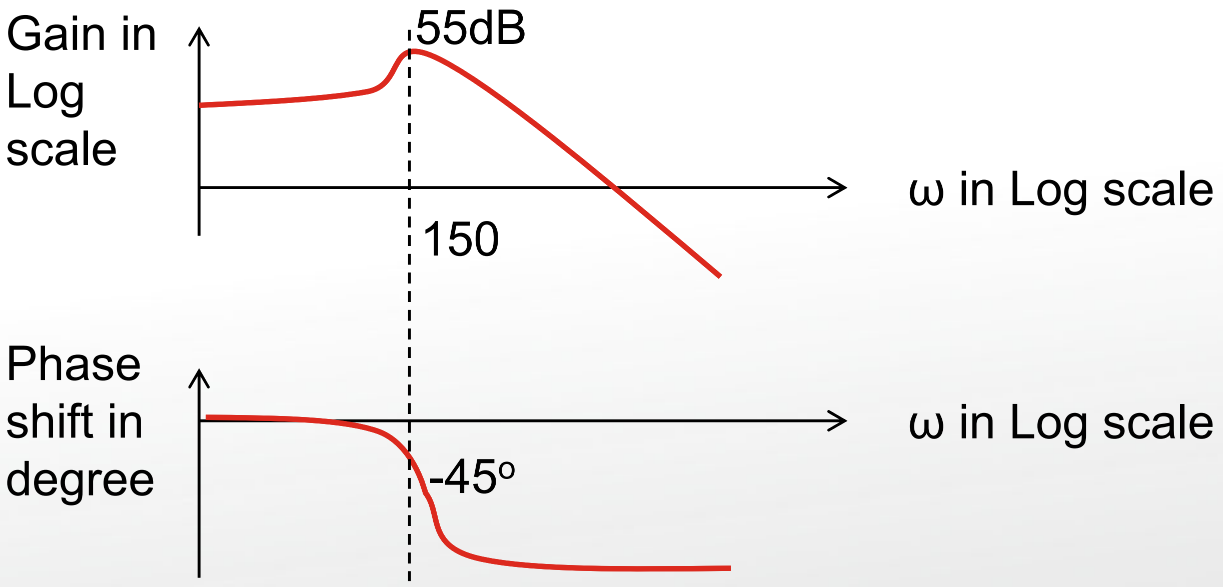
Q1: What is Bode Plot

- The interpretation of the Bode Plot can be understood using the following examples:
    - If the input sine wave has frequency 20rad/sec, then the output sine wave has an amplitude which is amplified by 40dB. The output sine wave is also lagging by close to 0 degree.



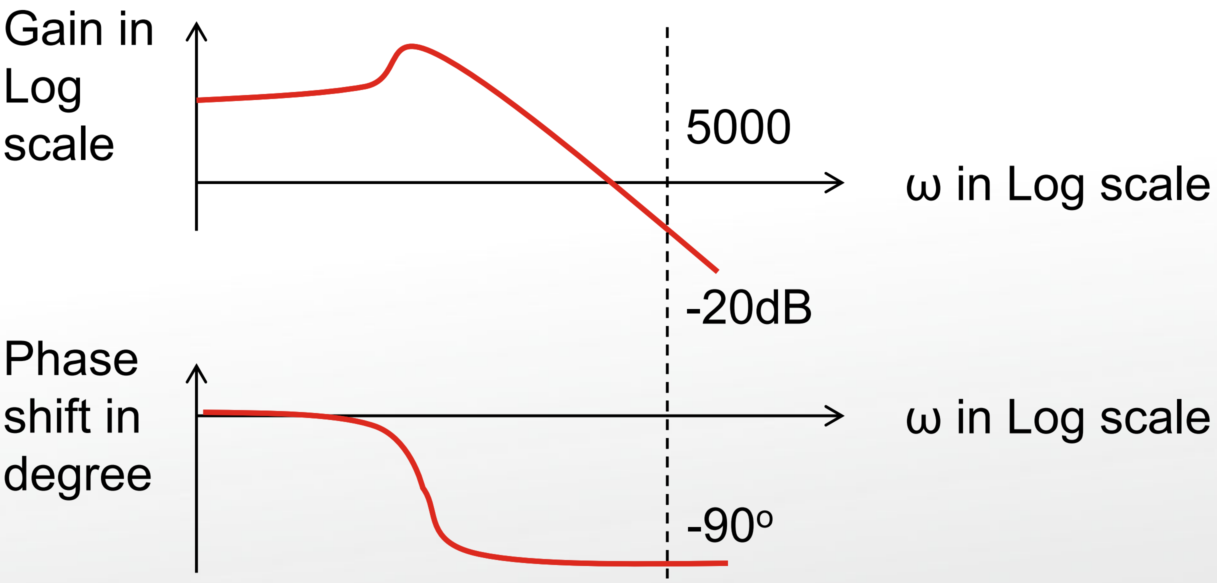
Q1: What is Bode Plot

- The interpretation of the Bode Plot can be understood using the following examples:
  - If the input sine wave has frequency 150rad/sec, then the output sine wave has an amplitude which is amplified by 55dB. The output sine wave is also lagging by close to 45 degree.



Q1: What is Bode Plot

- The interpretation of the Bode Plot can be understood using the following examples:
  - If the input sine wave has frequency 5000rad/sec, then the output sine wave has an amplitude which is amplified by -20dB (actually, this means smaller). The output sine wave is also lagging by close to 90 degree.



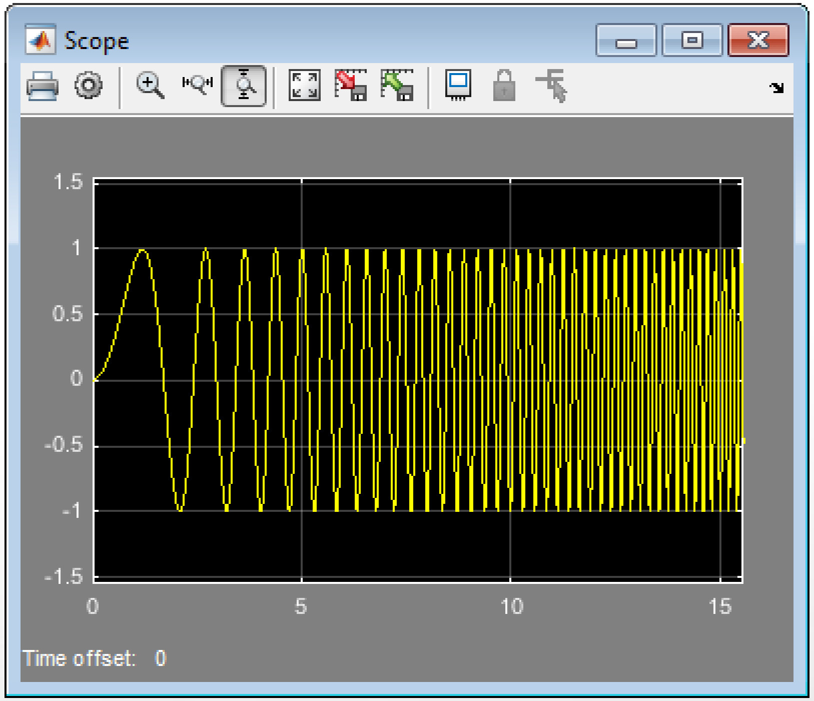
Q2: How to Obtain Bode Plot

- There are a few methods available:
  - The manual method:
    - Create sinusoidal excitation at a certain frequency.
    - Find the relative amplitude of output with respect to the input amplitude.
    - Find phase different between input and output.
    - Repeat the above steps for all frequencies in the range of interest.

➡ Tedious and troublesome.

Q2: How to Obtain Bode Plot

- The automated method:
  - We excite the system with a sine-sweep signal as input, i.e. a sinusoidal signal with increasing frequency.



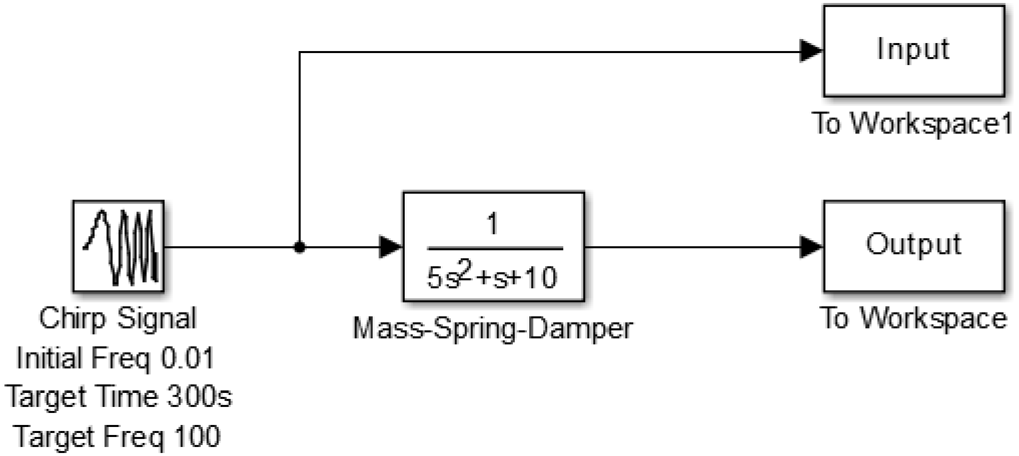
Q2: How to Obtain Bode Plot

- Then we collect both the input and output data.
- The frequency response can then be calculated by dividing the discrete Fourier Transform (DFT) of the output by the DFT of input:

$$\hat{G}(\omega) = \frac{\text{DFT of output}}{\text{DFT of input}}$$

MATLAB Simulink Example

- Create the following model in Simulink:

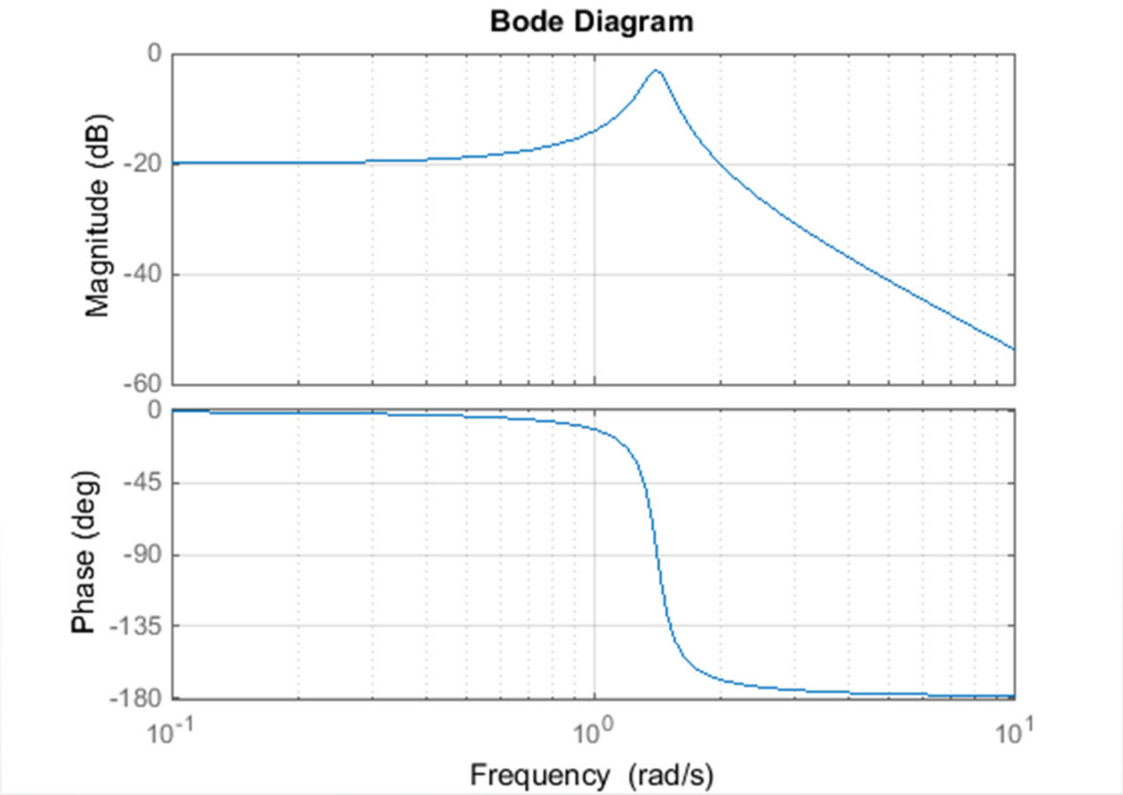


- Assume that we do not know the mass-spring-system, even though for simulation purpose we have to make up one.
- As usual, the configuration parameters are set as:
  - Fixed step size 1e-3 & Runge Kutta ode4
- Run the simulation for 300 seconds, and collect the input-output data.
  - In actual experiments, you could use LabVIEW to acquire the data.

MATLAB Simulink Example

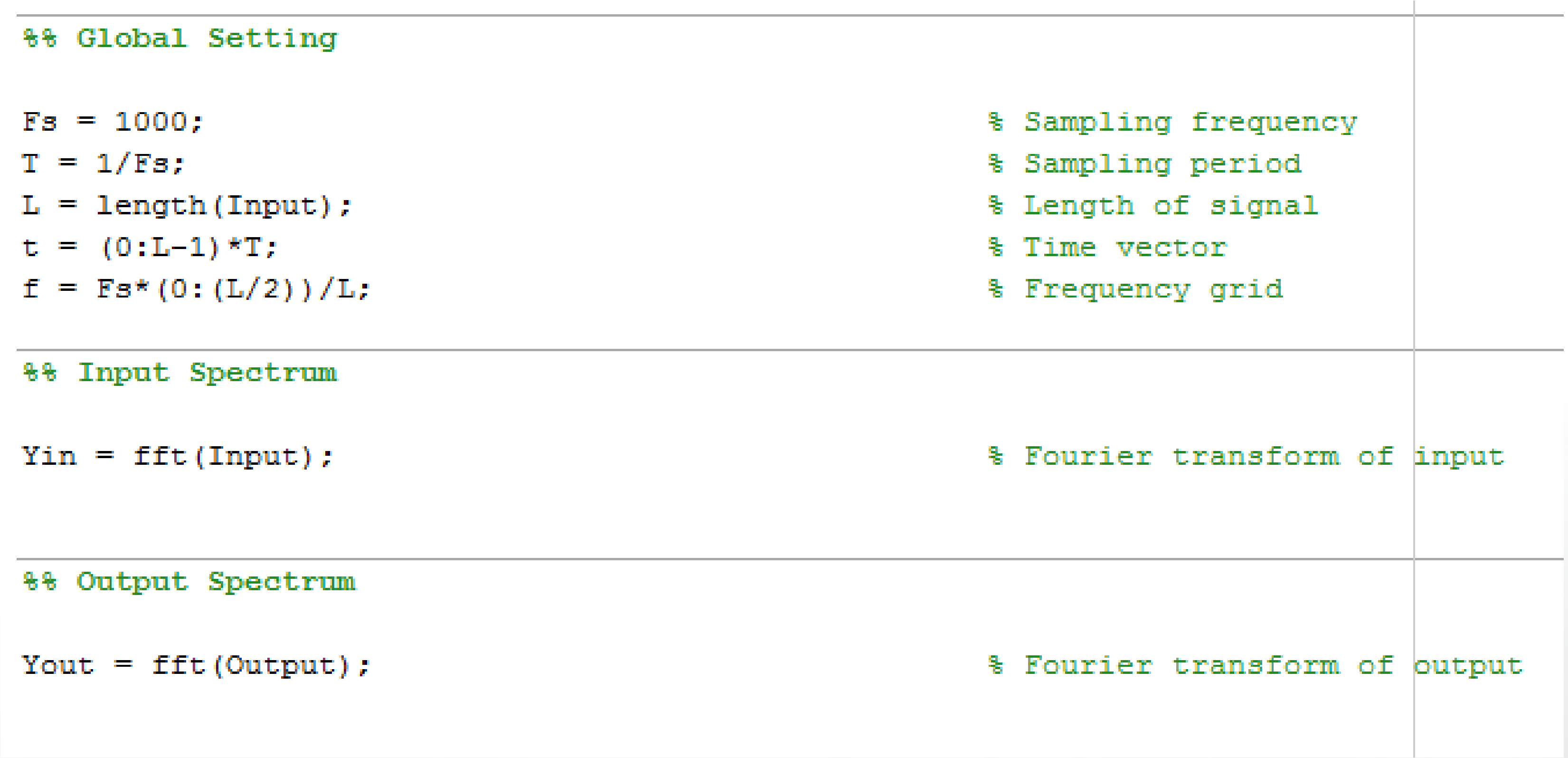
- Shown is the bode plot of the mass-spring-damper system, which is drawn using the MATLAB function “Bode”

```
s = tf(‘s’);
G = 1/(5*s^2 + 1*s + 10);
bode(G);
```



MATLAB Simulink Example

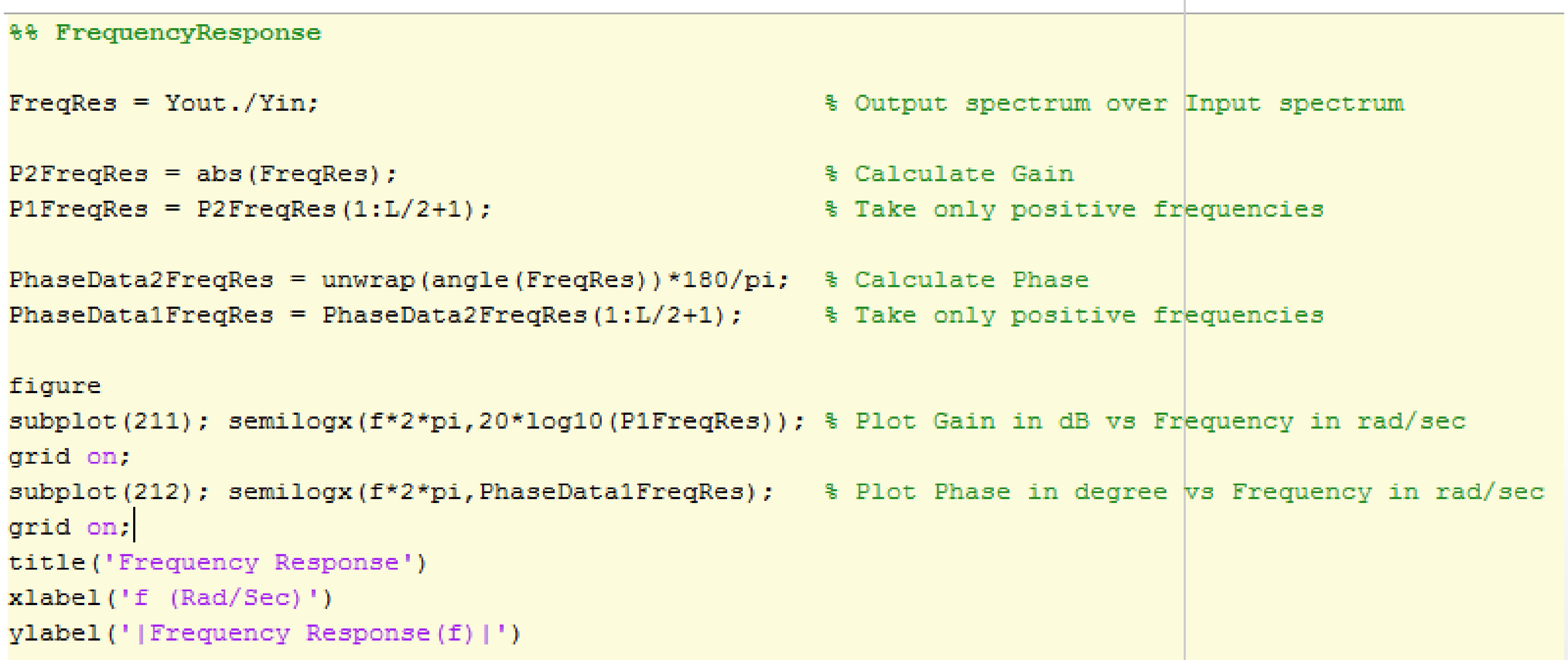
- To use Fourier transform to calculate the frequency response, write the following code in MATLAB:



```
%% Global Setting

Fs = 1000;                                          % Sampling frequency
T = 1/Fs;                                           % Sampling period
L = length(Input);                                  % Length of signal
t = (0:L-1)*T;                                      % Time vector
f = Fs*(0:(L/2))/L;                                 % Frequency grid

%% Input Spectrum

Yin = fft(Input);                                   % Fourier transform of input


%% Output Spectrum

Yout = fft(Output);                                 % Fourier transform of output
```

MATLAB Simulink Example

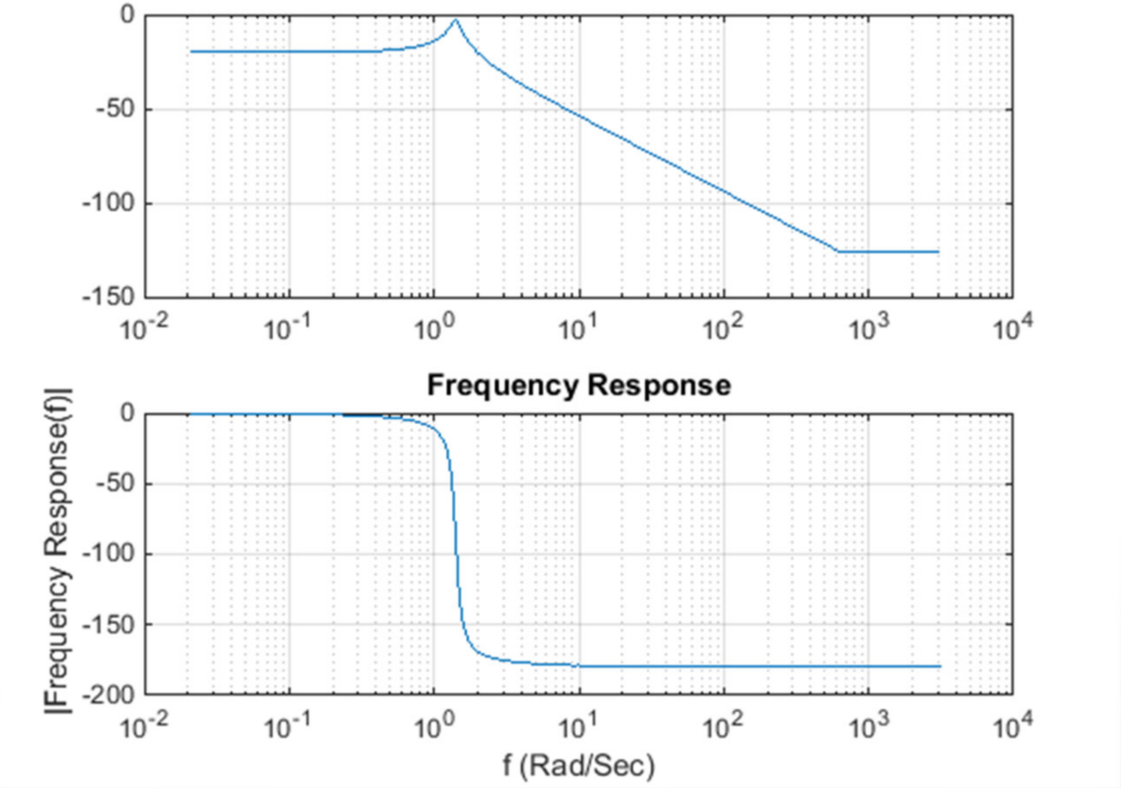
- Continued



```
%% FrequencyResponse

FreqRes = Yout./Yin;                                % Output spectrum over Input spectrum

P2FreqRes = abs(FreqRes);                           % Calculate Gain
P1FreqRes = P2FreqRes(1:L/2+1);                     % Take only positive frequencies

PhaseData2FreqRes = unwrap(angle(FreqRes))*180/pi;  % Calculate Phase
PhaseData1FreqRes = PhaseData2FreqRes(1:L/2+1);     % Take only positive frequencies

figure
subplot(211); semilogx(f*2*pi,20*log10(P1FreqRes)); % Plot Gain in dB vs Frequency in rad/sec
grid on;
subplot(212); semilogx(f*2*pi,PhaseData1FreqRes);   % Plot Phase in degree vs Frequency in rad/sec
grid on;
title('Frequency Response')
xlabel('f (Rad/Sec)')
ylabel('|Frequency Response(f)|')
```

MATLAB Simulink Example

- By clicking the run button, the following plot is obtained:



- And it looks the same as given by the “Bode” function up to 100Hz or 628 rad/sec.

Q3: How to Fit a Transfer Function

- Now, after we have obtained the Bode plot, how do we estimate the transfer function $\hat{G}(s) = \frac{N(s)}{D(s)}$?

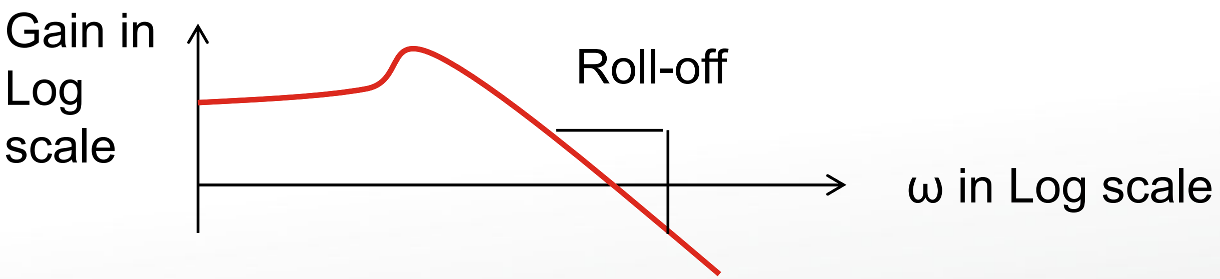
- The rigorous method:
  - Parameters of the transfer function model, i.e. The coefficients of N(s) and D(s) can be obtained by solving a nonlinear least square optimization problem that minimizes the cost function:

$$C = \frac{1}{n+1} \sum_{i=0}^{n} \left(G(j\omega_i) - \hat{G}(j\omega_i) \right)^2$$

  - Iterative methods such as gradient search algorithm can be used to solve this problem.

Q3: How to Fit a Transfer Function

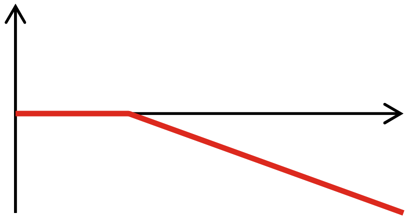
- The manual method:
  - We can first guess the relative order of N(s) and D(s) by the roll-off of the Bode Plot.
    - If the roll-off is -20dB/decade, then the relative order is one.
    - If the roll-off is -40dB/decade, then the relative order is two.



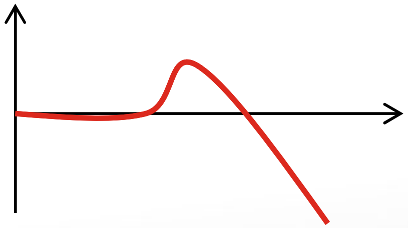
  - We can also guess the relative order by the phase change.
    - Every 90deg lag corresponds to one relative order.

Q3: How to Fit a Transfer Function

- Next, we start off by getting a constant DC gain of the system by reading this value from the Bode Diagram.
- Then, we slowly fit the frequency response by multiplying several 1st or 2nd order transfer functions, for which we know their bode plots:



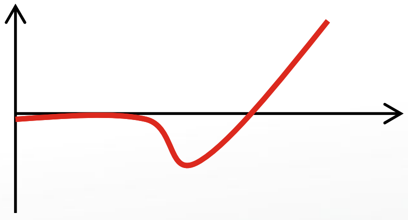
$$\frac{1}{Ts+1}$$



$$\frac{\omega^2}{s^2+2\xi\omega s+\omega^2}$$

Resonance

(Peak)



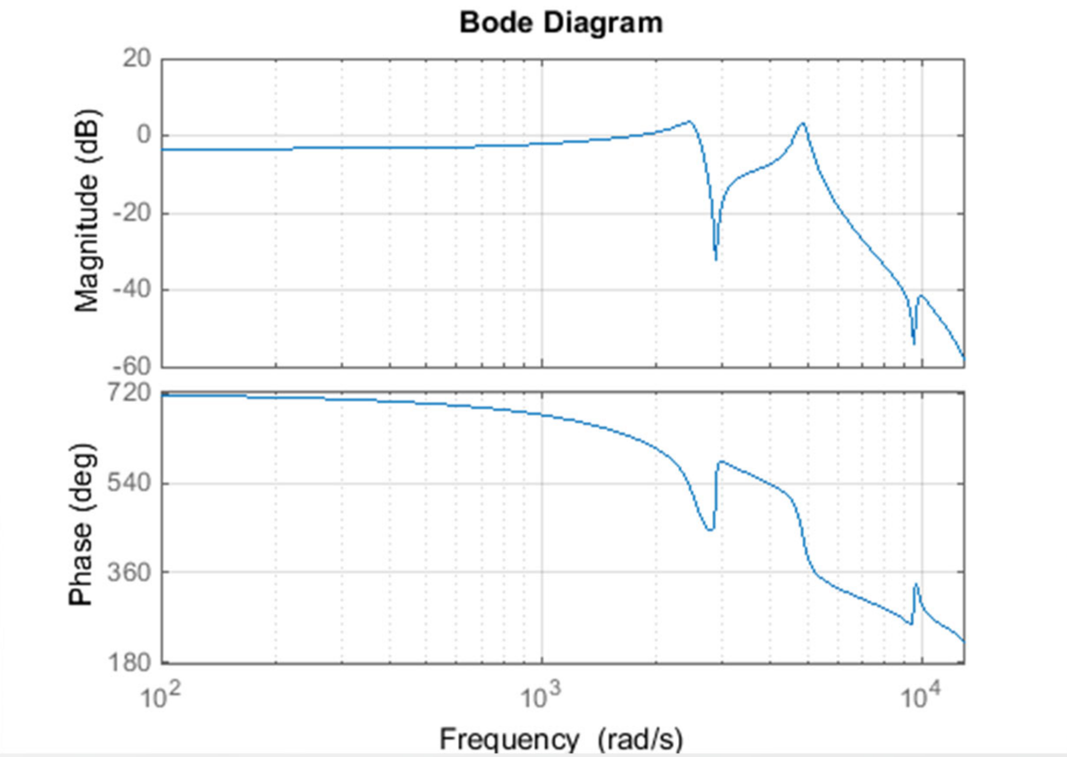
$$\frac{s^2+2\xi\omega s+\omega^2}{\omega^2}$$

Anti-Resonance

(Dip)

Matlab Example

- Example: The following is the frequency response of a Piezoactuator, obtained through measurement.



Matlab Example

- Let’s fit a transfer function model to the frequency response.
- Firstly, the DC gain of the frequency response plot is -3.44dB.
- Therefore the DC gain value can be calculated as:

$$
\begin{aligned}
&20\log_{10} x = -3.44 \\
&\log_{10} x = -0.172 \\
&x = 10^{-0.172} = 0.673
\end{aligned}
$$

- We thus start building the model as:

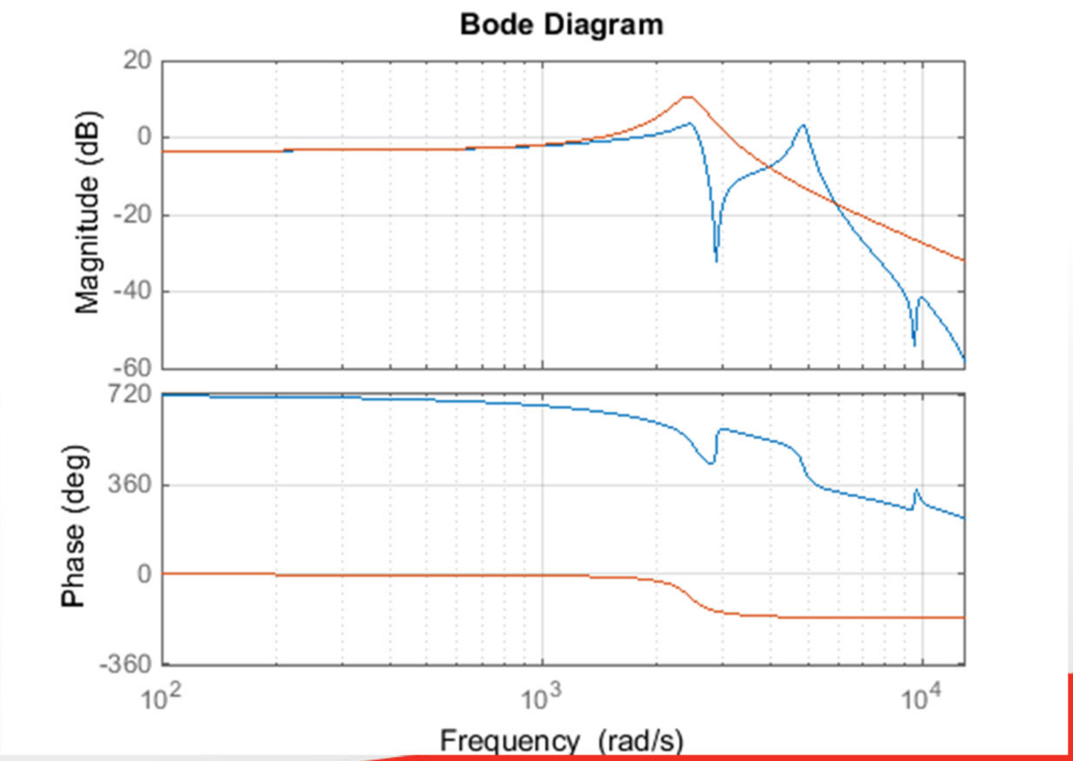
$$\hat{G}(s) = 0.673$$

Matlab Example

- Next, we look at the frequency response and notice that there is a peak at around 2450 rad/s.
- Therefore we extend the model to:

$$\hat{G}(s) = 0.673 \cdot \frac{2450^2}{s^2 + 2(0.1)2450s + 2450^2}$$

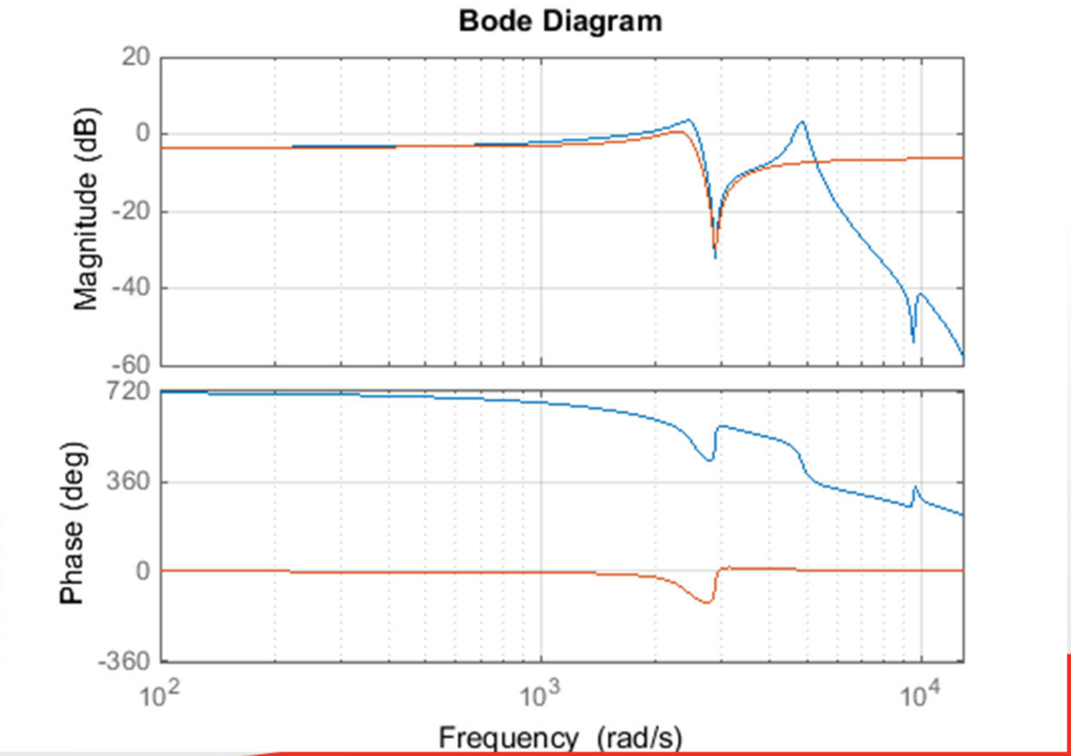
  - Where the damping ratio 0.1 is set by fine tuning.



Matlab Example

- Now, there is a "dip" at around 2875 rad/sec.
- Therefore the model is updated to:

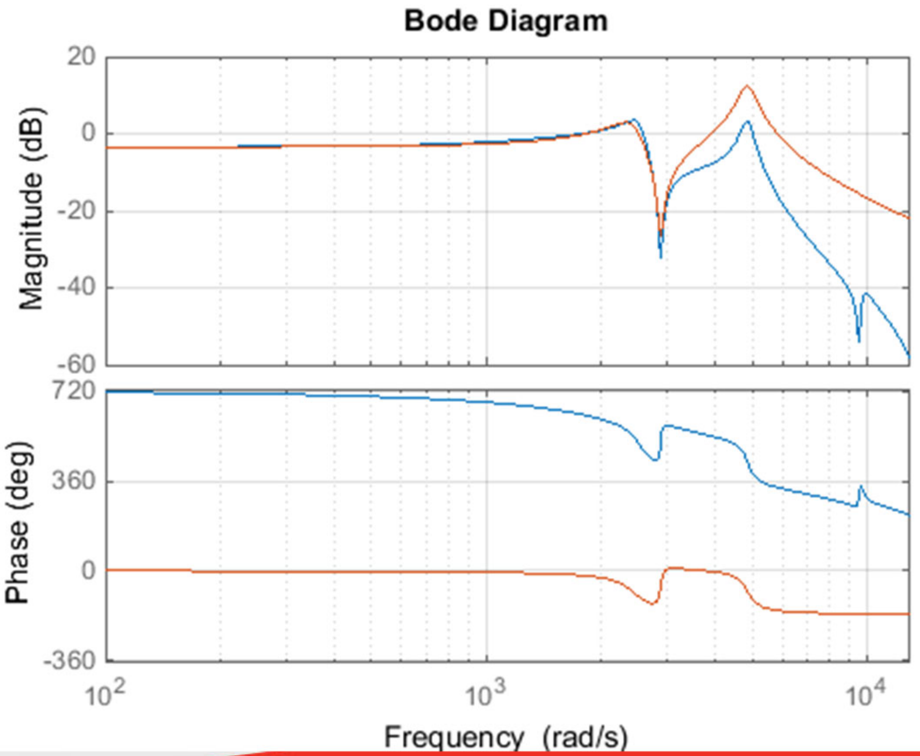
$$\hat{G}(s) = 0.673 \cdot \frac{2450^2}{s^2 + 2(0.1)2450s + 2450^2} \cdot \frac{s^2 + 2(0.01)2875s + 2875^2}{2875^2}$$



Matlab Example

- The next peak at around 4860 rad/sec is slightly more challenging to fit.
- First, the model is extended to:

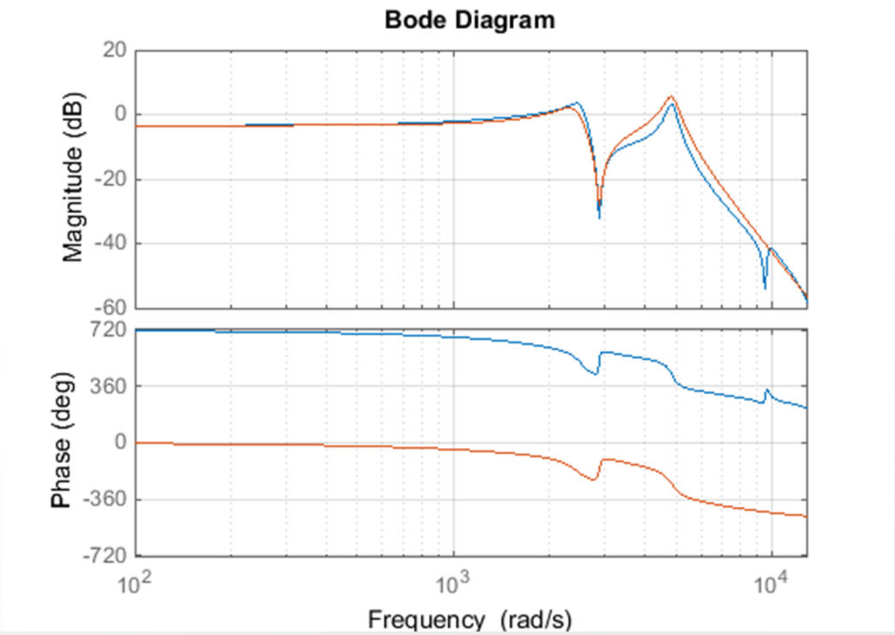
$$\hat{G}(s) = 0.673 \cdot \frac{2450^2}{s^2 + 2(0.1)2450s + 2450^2} \cdot \frac{s^2 + 2(0.01)2875s + 2875^2}{2875^2} \cdot \frac{4860^2}{s^2 + 2(0.05)4860s + 4860^2}$$



Matlab Example

- It is noticed that the phase didn't drop as much as it should. Therefore more resonance modes are added to 4860 rad/sec with different damping.

$$\hat{G}(s) = 0.673 \cdot \frac{2450^2}{s^2 + 2(0.1)2450s + 2450^2} \cdot \frac{s^2 + 2(0.01)2875s + 2875^2}{2875^2} \cdot \frac{4860^2}{s^2 + 2(0.05)4860s + 4860^2}$$
$$\cdot \frac{4860^2}{s^2 + 2(0.7)4860s + 4860^2} \cdot \frac{4860^2}{s^2 + 2(0.8)4860s + 4860^2}$$



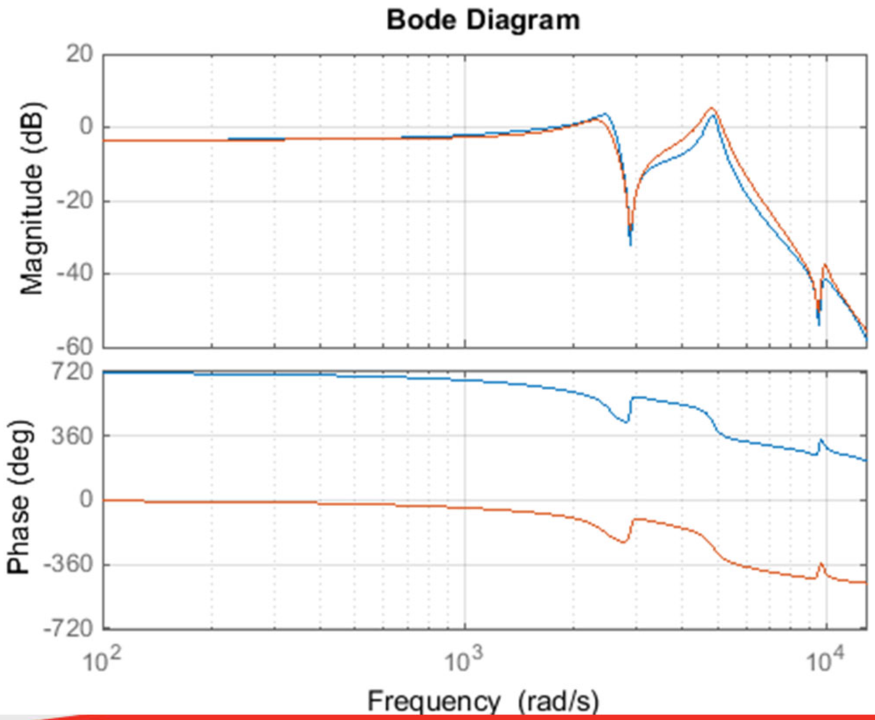
- At this stage, the fitted model is good enough if we know we will excite the system up to around 8000 rad/sec.
- If we wish to further capture the remaining dip and peak at around 9000 rad/sec, we can extend the model to the form as shown in next slide:

Matlab Example

$$\hat{G}(s) = 0.673 \cdot \frac{2450^2}{s^2 + 2(0.1)2450s + 2450^2} \cdot \frac{s^2 + 2(0.01)2875s + 2875^2}{2875^2} \cdot \frac{4860^2}{s^2 + 2(0.05)4860s + 4860^2}$$

$$\cdot \frac{4860^2}{s^2 + 2(0.7)4860s + 4860^2} \cdot \frac{4860^2}{s^2 + 2(0.8)4860s + 4860^2}$$

$$\cdot \frac{s^2 + 2(0.01)9540s + 9540^2}{9540^2} \cdot \frac{9800^2}{s^2 + 2(0.02)9800s + 9800^2}$$



Reference

- Lecture Notes on “Modelling of Mechatronics System”, Abdullah Al-Mamun, National University of Singapore.
- Lecture Notes: “Mechatronics Design”, Dr Milan Simic, RMIT University

Thank you, Questions