System Identification in Frequency-Domain

- **Objective**
  - Building dynamic model of a *stable* LTI system based on experimentally obtained frequency responses of the system
- **General Procedure and Common Issues**
  - Experimental design and frequency experimental tests
  - Data processing to obtain frequency responses
  - Transfer function determination to fit the obtained frequency responses
  - Model validation
- **Parameter Estimation for LTI Systems via Frequency Responses**
  - First-order Systems
  - Second-order Systems

Experimental Design

- **Identifiability Requirement**

  Input signals used for experimental tests need to be rich enough to cover entire frequency-range of interests, which can be done by

  - (i) multiple tests with sinusoidal inputs of single frequency spanning the range, or
  - (ii) single test with a rich complicated input (e.g., periodic input of multi-sinusoids)

- **Minimizing Model Error Effect**

  - Choose input signals large enough to have good signal-to-noise ratio and to reduce the effect of unavoidable nonlinear elements in experiments (e.g., Coulomb friction), while not being overly large that they induce additional model error by violating the underline linear assumptions (e.g., input and state saturations due to too large input signals).

Experiments and Data Processing

- **Steady-State Response Requirement**

  Conduct experiments with the test input signals long enough to make sure that the recorded measurement responses do converge to their *steady-state* responses!

- **Experimental Tests with Sinusoids of Single Frequency**

  - At each frequency ω_i of the sinusoidal inputs, obtain the magnitude ratio M_i and phase shaft ϕ_i of the steady-state sinusoidal response $y_{ss} = A_y \sin\left(\omega_i t + \phi_i\right)$ to the input $u = A_u \sin\left(\omega_i t\right)$ to get $G\left(j\omega_i\right)$:

$$\begin{aligned} \left|G\left(j\omega_i\right)\right| &= M_i \\ \angle G\left(j\omega_i\right) &= \phi_i \end{aligned} \quad \Rightarrow \quad \begin{cases} \left\|G\left(j\omega_i\right)\right\| = 20\log_{10}\left(M_i\right) \\ \angle G\left(j\omega_i\right) = \phi_i \times \dfrac{180^0}{\pi} \end{cases}$$

where

$$M_i = \frac{A_y}{A_u}, \quad \phi_i = \omega_i \Delta t$$

Frequency Response via System ID Tool Box

- **Experimental Test**
  - Does not need to perform many sinusoidal tests at different frequencies. Instead, generate a periodic input $u(t)$ that contains components of all interested frequencies. Apply this periodic input and obtain the *steady-state* periodic response $y(t)$ via one experimental test.

- **Data Processing**
  - Obtain the Fourier Transform of input, $U(j\omega)$, and the steady-state output, $Y(j\omega)$, respectively

  - Obtain Frequency Response by

$$G(j\omega) = \frac{Y(j\omega)}{U(j\omega)}$$

MATLAB System ID Tool Box

- **Input Signal Generation**

Freq. Resolution: $\frac{1}{N} f_s$ Hz

Data Length: $N \frac{1}{f_s}$ sec

>> help idinput

IDINPUT Generates input signals for identification.

U = IDINPUT(N,TYPE,BAND,LEVELS)

U: The generated input signal. A column vector or a N-by-nu matrix.

N: The length of the input.

N = [N Nu] gives a N-by-Nu input (Nu input channels).

N = [P Nu M] gives a M*P-by-Nu input, periodic with period P and with M periods.

Default values are Nu = 1 and M =1 ;

TYPE: One of the following:

'RGS': Generates a Random, Gaussian Signal.

'RBS': Generates a Random, Binary Signal.

'PRBS': Generates a Pseudo-random, Binary Signal.

'SINE': Generates a sum-of-sinusoid signal.

Default: TYPE = 'RBS'.

BAND: A 1 by 2 row vector that defines the frequency band for the input's frequency contents.

For the 'RS', 'RBS' and 'SINE' cases BAND = [LFR,HFR], where LFR and HFR are the lower and upper limits of the passband, expressed in fractions of the Nyquist frequency (thus always numbers between 0 and 1).

For the 'PRBS' case BAND = [0,B], where B is such that the signal is constant over intervals of length 1/B (the Clock Period).

Default: BAND =[0 1].

LEVELS = [MI, MA]: A 2 by 1 row vector, defining the input levels.

For 'RBS', 'PRBS', and 'SINE', the levels are adjusted so that the input signal always is between MI and MA.

For the 'RGS' case, MI is the signal's mean value minus one standard deviation and MA is the signal's mean plus one standard deviation.

Default LEVELS = [-1 1].

In the 'PRBS' case, if M > 1, the length of the data sequence and the period is adjusted so that always an integer number of maximum length PRBS periods are obtained. If M = 1 the period is chosen so that it becomes longer than P = N. In the multiinput case the signals are maximally shifted. This means that P/Nu is an upper bound for the model orders that can be used to identify systems excited by such a signal.

In the 'SINE' case, the sinusoids are chosen from the frequency grid freq = 2*pi*[1:Grid_Skip:fix(P/2)]/P intersected with pi*[BAND(1) BAND(2)]. (for Grid_Skip see below.) For multi-input signals, the different inputs use different frequencies from this grid. An integer number of full periods is always delivered. The selected frequencies are obtained as [U,FREQS] = IDINPUT(....), where row ku of FREQS contains the frequencies of input number ku. The resulting signal is affected by a 5th input argument SINEDATA:

U = IDINPUT(N,TYPE,BAND,LEVELS,SINEDATA)

where

SINEDATA= [No_of_Sinusoids, No_of_Trials, Grid_Skip],

meaning that No_of_Sinusoids are equally spread over the indicated BAND, trying No_of_Trials different, random, relative phases, until the lowest amplitude signal is found.

Default SINEDATA = [10,10,1];

See also idmodel/sim.

$$\sum_{i=1}^{N} \sin(\omega_i t + \phi_i)$$

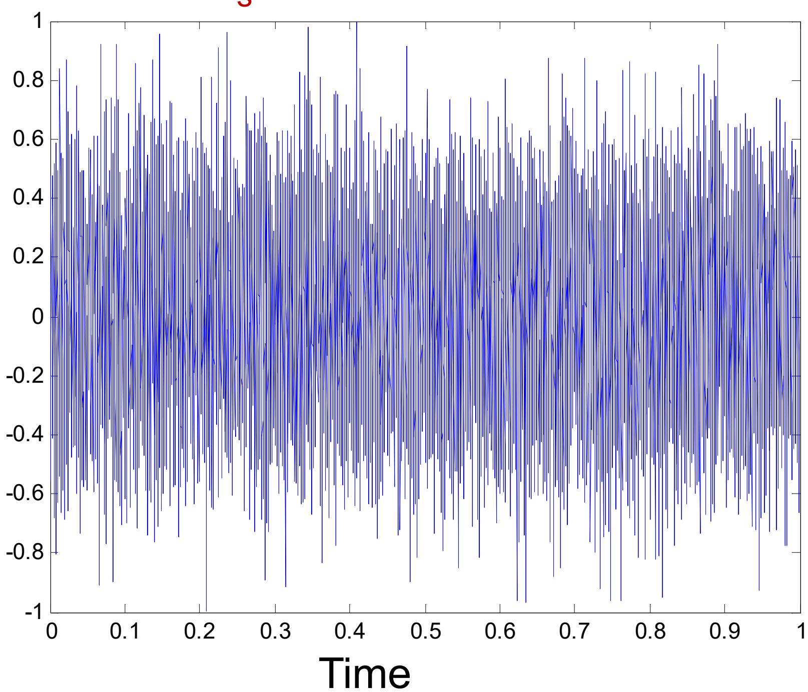
"***idinput***" command to generate a sum of 1000 sinusoids ranging from 1Hz to 5KHz with a sampling frequency of 10KHz:

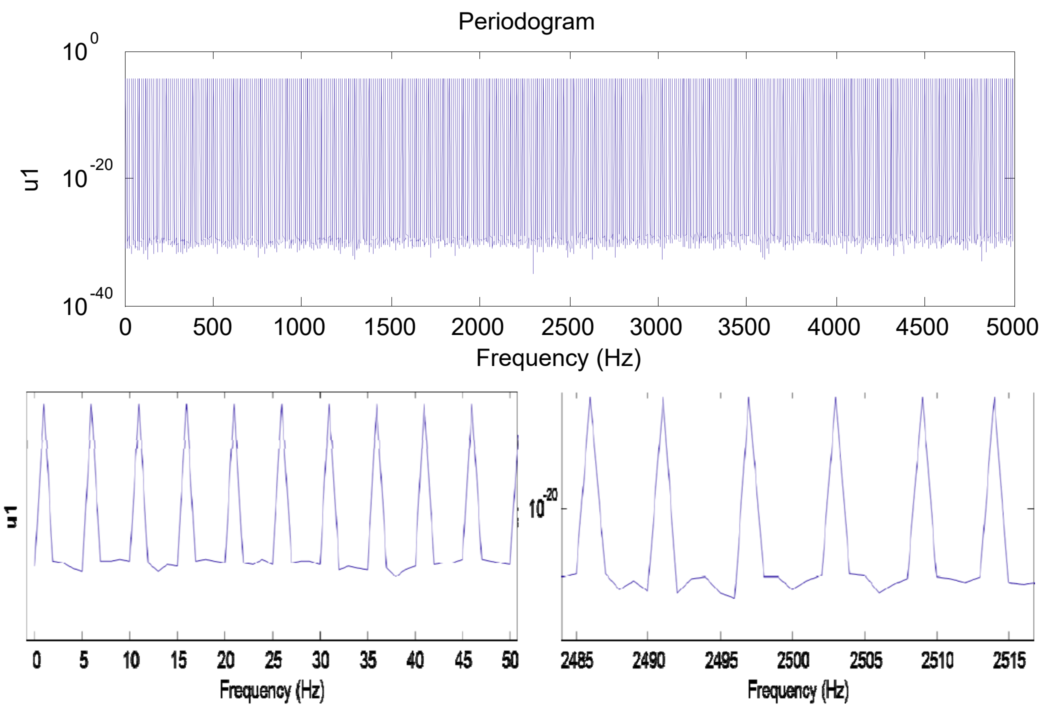
```
sinesum = idinput([10000 1 1],'sine',
                          [0 1],[-1 1],[1000 50 1]);
```

Freq. Resolution: $\frac{1}{N} f_s$ Hz

Data Length: $N \frac{1}{f_s}$ sec

Use f_s=10KHz

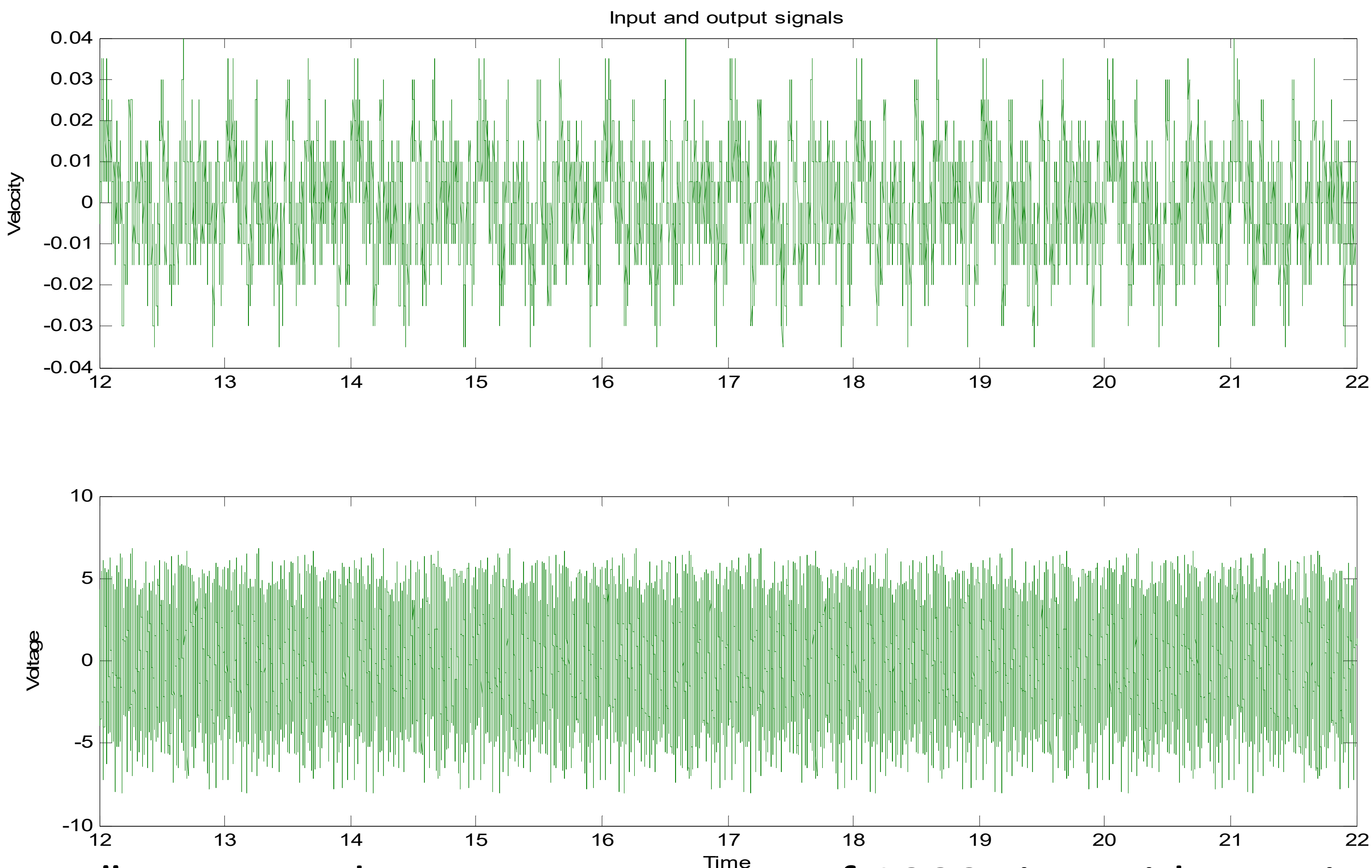




499 sinusoids from 1Hz to 2491Hz and 499 sinusoids from 2509Hz to 4999Hz with 5Hz spacing, 2 sinusoids at 2497Hz and 2503Hz with 6Hz spacing

PURDUE MECHANICAL ENGINEERING

System Dynamics Identification – Single Sinusoidal Test

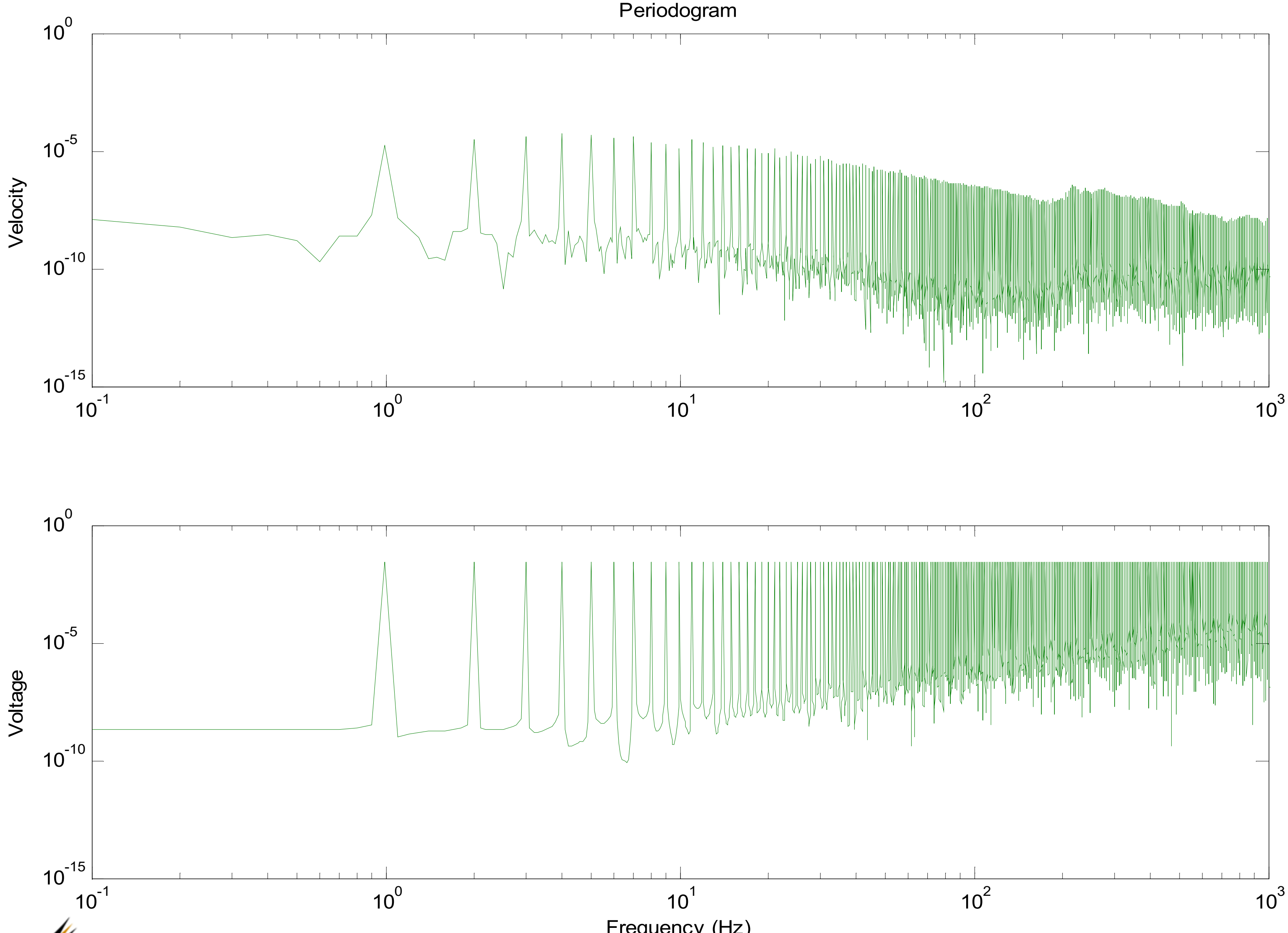


"***idinput***" command to generate a sum of 1000 sinusoids ranging from 1Hz to 1KHz with a sampling frequency of 10KHz:

```
sinesum = idinput([10000 1 1],'sine',
                          [0 0.2],[-1 1],[1000 50 1]);
```

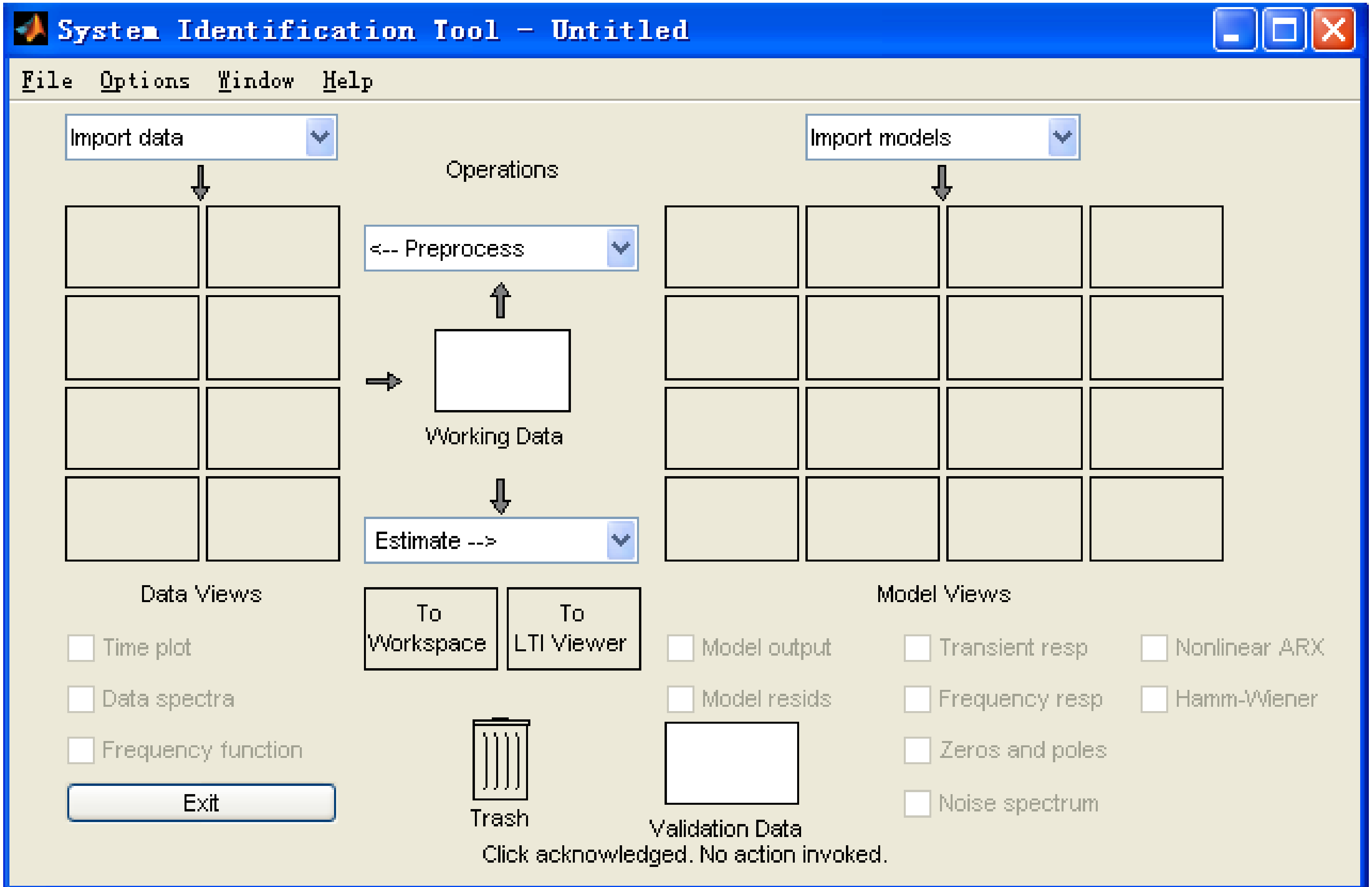
PURDUE MECHANICAL ENGINEERING

System Dynamics Identification – Single Sinusoidal Test



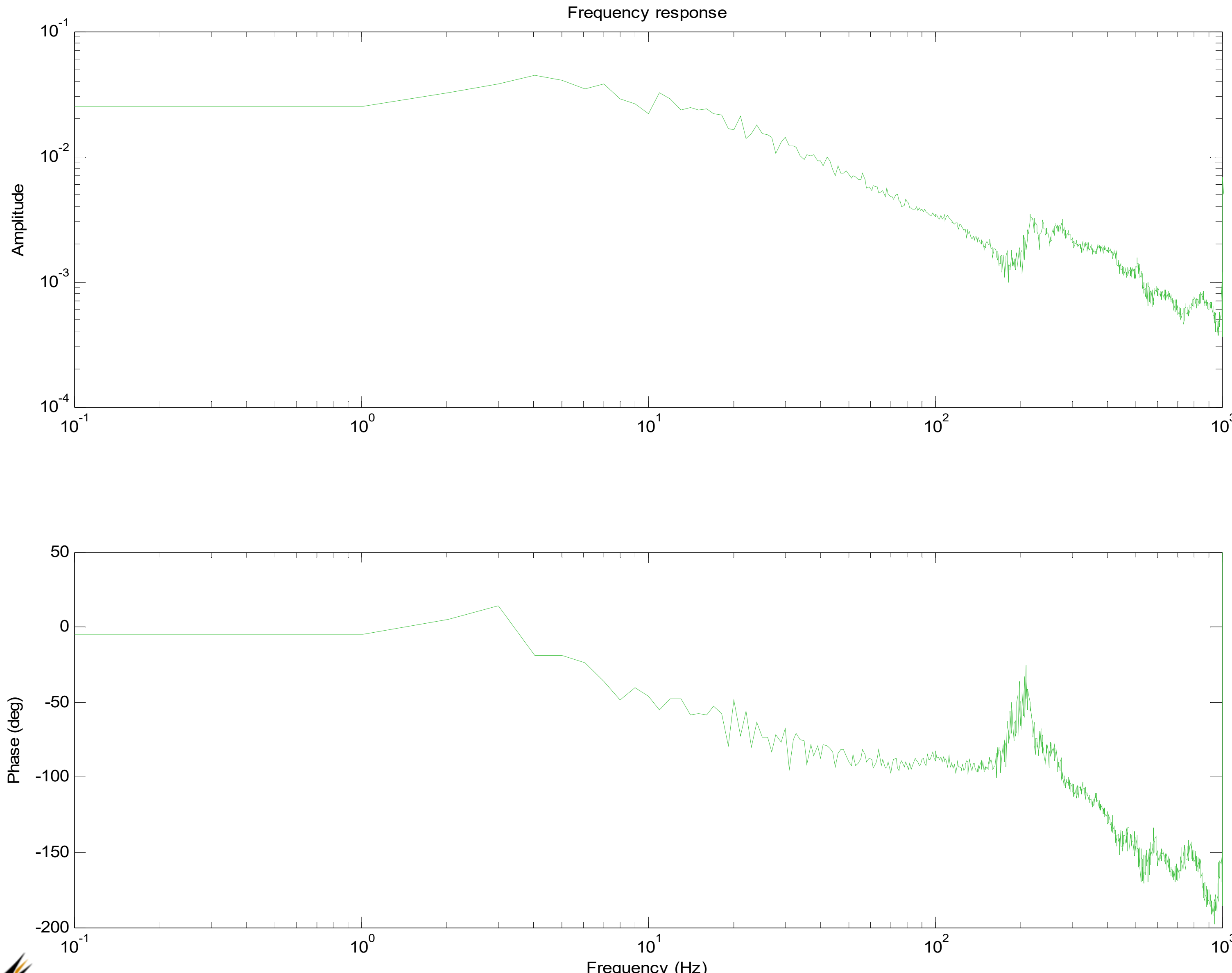
Spectrum of Input/Output using MATLAB "***fft***"

System Identification Tool Box GUI



ident

System Dynamics Identification – Single Sinusoidal Test



Frequency Response Estimate using MATLAB "***SPFDR***" with 5000f

PURDUE MECHANICAL ENGINEERING

Determine Transfer Function via Bode Plots

- Use straight-lines of ***exact*** slopes of integer multiplications of ± 20 dB/decade to fit the Bode magnitude plot.
- Start from both low-frequency and high-frequency with accurate straight-line fitting. Then work toward middle based on slope changes of the Bode magnitude plot. Keep in mind that
  - (i) around 3dB error at the break-frequency of a first-order pole/zero,
  - (ii) large local peak/valley close to the break-frequency of lightly damped second-order poles/zeros, and
  - (iii) no peak/valley for $0.707 < \zeta < 1$ but no more than 6dB error at the break-frequency of an underdamped second-order poles/zeros with $0.707 < \zeta < 1$
- Write down the form of the transfer function -- the straight-line approximations give all the information needed to determine the low-frequency gain, the number of integrators/differentiators, the order and break-frequencies of stable poles and MP or NMP zeros. Use Bode phase plot to determine if the zero is MP or NMP.

Determine Transfer Function via Bode Plots

– To determine the damping ratio of a second-order pole/zero,

(i) For lightly damped poles/zeros, obtain the resonance peak value by locating the maximal discrepancy of the Bode magnitude plot and straight-line approximations around the break-frequency. Use the resonance peak formula of

$$\left|G_{p2}(j\omega_r)\right| = \frac{1}{2\zeta\sqrt{1-\zeta^2}}$$

to back out the damping ratio

(ii) Alternatively, determine the discrepancy of Bode magnitude plot and straight-line approximations at the break-frequency. Use the break-frequency magnitude formula of

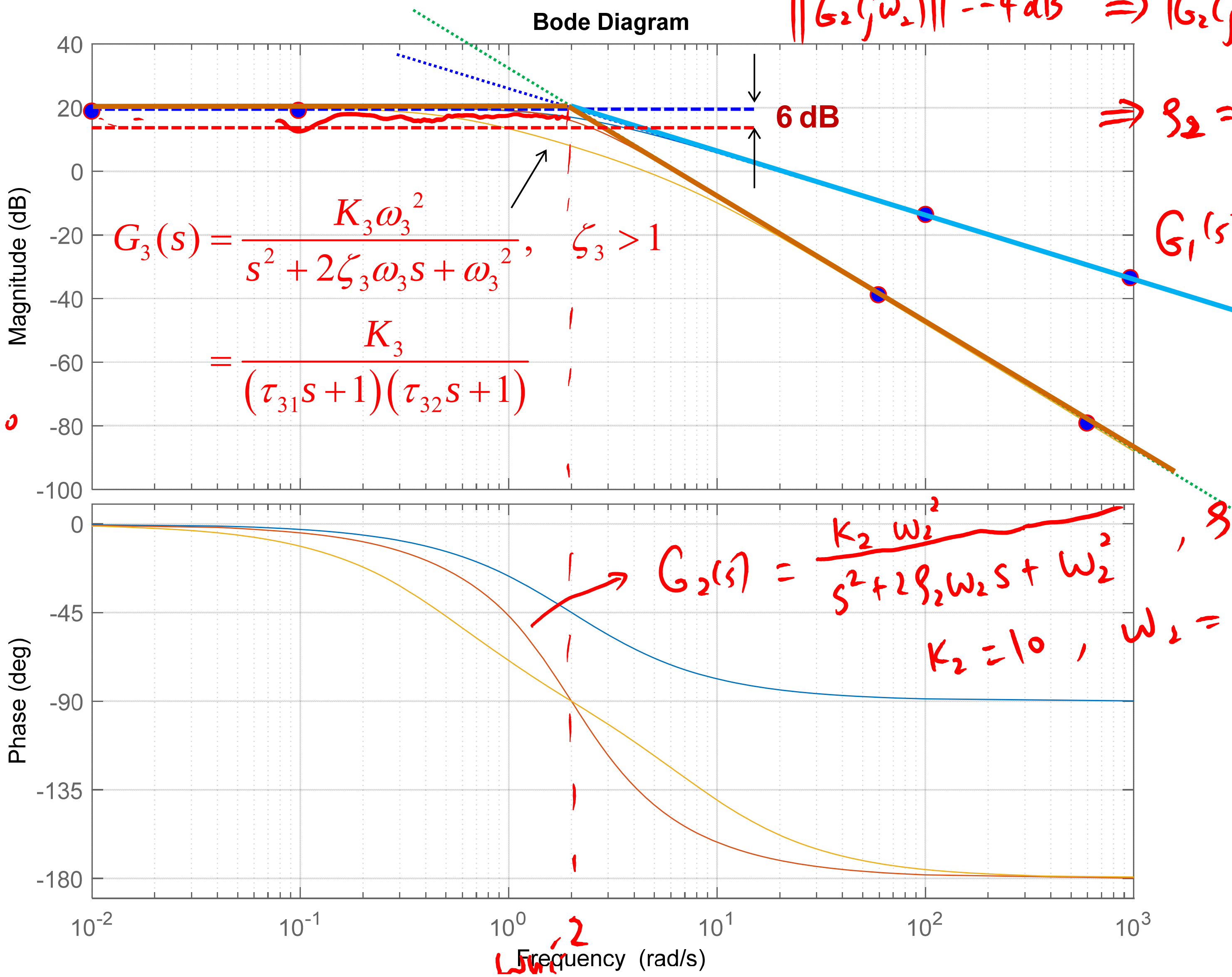
$$\left|G_{p2}(j\omega_n)\right| = \frac{1}{2\zeta}$$

to back out the damping ratio

– Finally, use the Bode phase plot to double check correctness of the transfer function (TF) you write down.

Example 1 – TF via Bode Plots

- Given the experimentally obtained frequency responses shown below, determine transfer functions that fit the obtained Bode plots



$\|G_2(j\omega_2)\| = -4\,dB \Rightarrow |G_2(j\omega_2)| = 0.63 = \frac{1}{2\zeta_2}$

$\Rightarrow \zeta_2 = 0.79$

$G_1(s) = \frac{K_1}{\tau_1 s+1}$

$K_1 = 10$

$\tau_1 = \frac{1}{\omega_{b1}} = \frac{1}{2}$

$K_3 = 10$

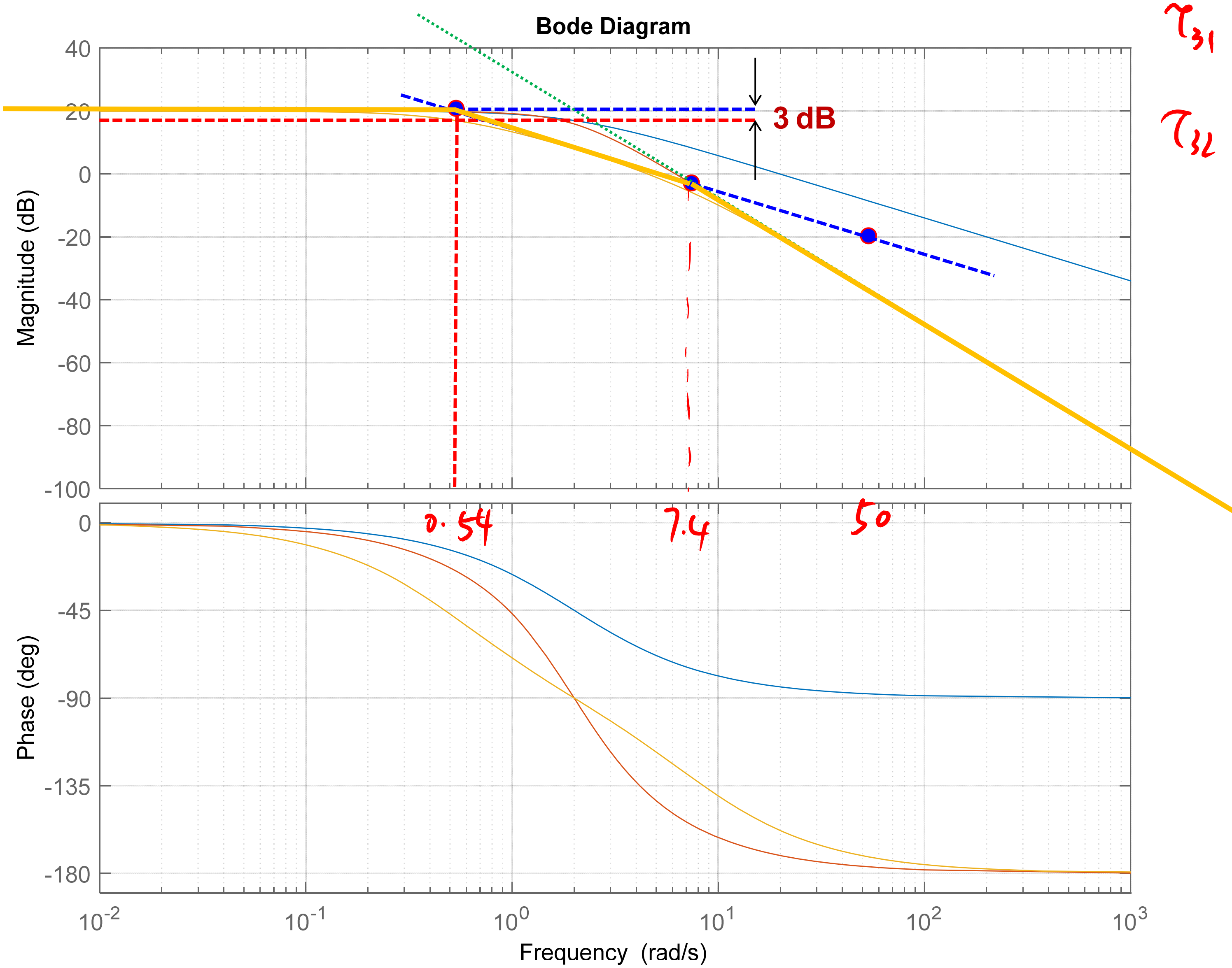
$G_2(s) = \frac{K_2\omega_2^2}{s^2+2\zeta_2\omega_2 s+\omega_2^2}, \quad \zeta_2 < 1$

$K_2 = 10, \quad \omega_2 = 2$

$\omega_{b1} = 2$

Example 1 – TF via Bode Plots

- Given the experimentally obtained frequency responses shown below, determine transfer functions that fit the obtained Bode plots

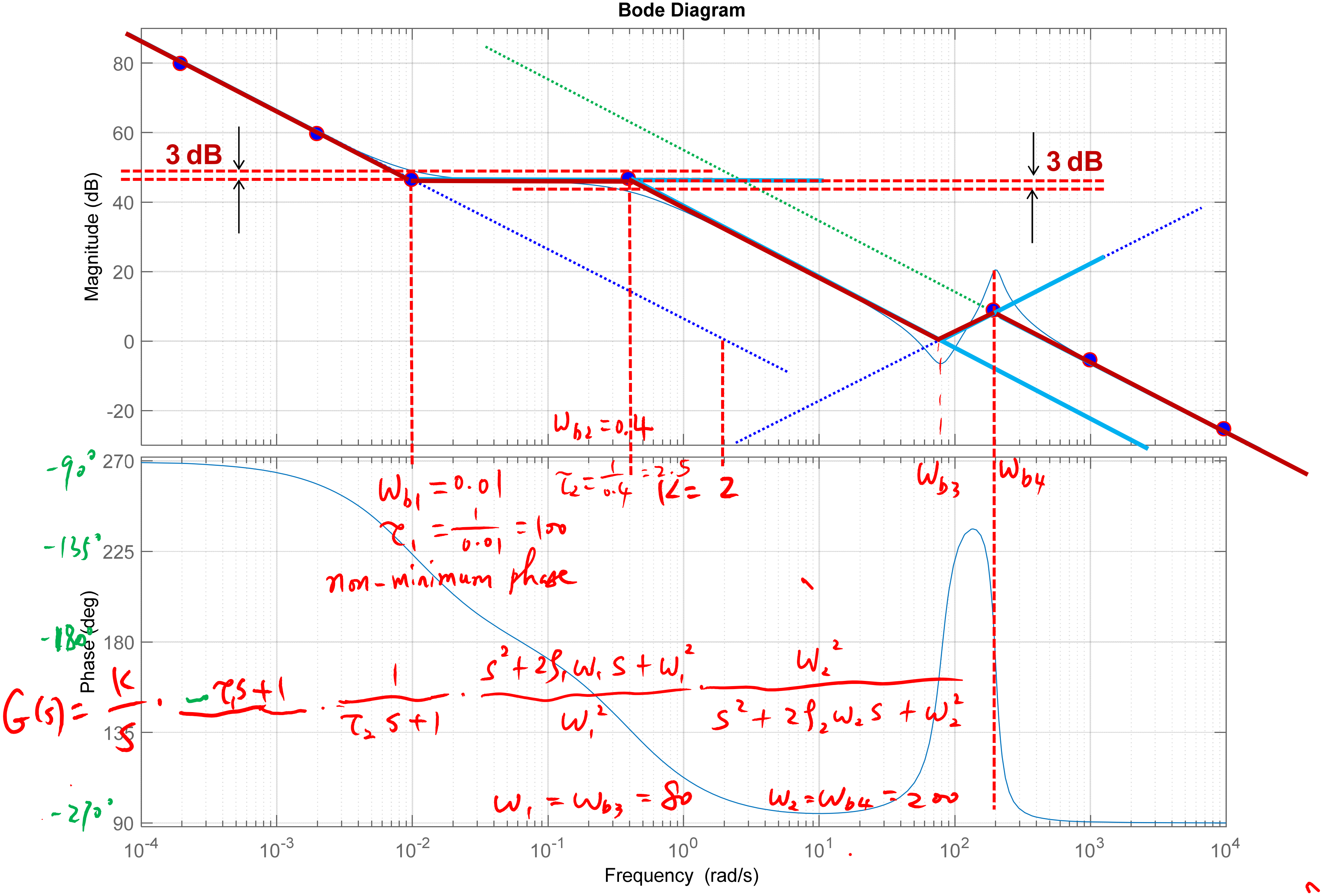


$\tau_{31} = \frac{1}{0.54}$

$\tau_{32} = \frac{1}{7.4}$

Example 2 – TF via Bode Plots

- Given the experimentally obtained frequency responses shown below, determine transfer functions that fit the obtained Bode plots



$$G(s) = \frac{K}{s} \cdot \frac{-\tau_1 s + 1}{} \cdot \frac{1}{\tau_2 s + 1} \cdot \frac{s^2 + 2\zeta_1 \omega_1 s + \omega_1^2}{\omega_1^2} \cdot \frac{\omega_2^2}{s^2 + 2\zeta_2 \omega_2 s + \omega_2^2}$$

Example 2 – TF via Bode Plots

- Given the experimentally obtained frequency responses shown below, determine transfer functions that fit the obtained Bode plots