An Undergraduate System Identification Experiment

Bradley T. Burchett and Richard A. Layton

Abstract—One of the essential objectives of any undergraduate controls curriculum is understanding the frequency response, its physical meaning, and stability consequences. In addition, the student should gain an appreciation for simple system identification techniques, and the relationship between frequency response and the system complex plane pole-zero map. These skills allow the student to analyze "black box" systems. With these objectives in mind, the authors have developed a system identification experiment using commercially available physical plants and digital computer controllers. This paper outlines the present course background, lesson objectives, data collection and reduction techniques for a system identification laboratory taught in the introductory system dynamics course in the Mechanical and Electrical Engineering programs at Rose-Hulman Institute of Technology.

I. INTRODUCTION

Rose-Hulman has a unique sophomore engineering curriculum that culminates in a five-credit hour course in system dynamics. This course, Analysis and Design of Engineering Systems (ADES) is a comprehensive overview of modeling, analysis, and simulation of mechanical, electrical, thermal and fluid systems and is taught to Electrical, Biomedical and Mechanical Engineering majors in an interdisciplinary setting. One of the mosst important objectives of this course is that the student learn to model systems using lumped parameters, and understand the assumptions and limitations of such modeling.

Frequency response is one of the fundamental concepts in linear system theory and classical control. We have found that our mechanical engineering students have a particularly difficult time relating frequency response to the physical world. This laboratory gives the student an opportunity to gather frequency response data, while watching the response of a physical system, summarize this data on a Bode plot, and determine the underlying Laplace domain and state space models. This is the eighth laboratory in the course.

II. LESSON OBJECTIVES

The objectives of this lab are as follows. Students will become familiar with the Educational Control Products (ECP) Rectilinear Control System [1], and Matlab [2] interface. They will collect input-output data for several sinusoidal inputs to the system. In doing so, they will learn

the physical meaning of frequency response. They will then compile their data on a Bode magnitude plot. This exercise will augment understanding of the Bode plot and spectral representations of frequency response data. Students will determine a fourth order transfer function model from the Bode magnitude plot. As shown in the sequel, the instructor may require the students to determine parametric transfer function, non-parametric transfer function, or parametric state space models. Since the phase data is not used, the student must infer stability and pole locations from the system physical characteristics. This emphasizes the relationship between frequency response plots and the system pole-zero map.

III. DATA COLLECTION

We developed this lab on the ECP Rectilinear Control System [1] which is shown in Figure 1. The system consists of the hardware plant which is essentially a three degree of freedom spring-mass-damper system, and a digital computer running ECP's own Interface Software "Executive Program". The ECP Interface Software provides an ASCII data output capability which allows for easy import of the data to Matlab for off-line analysis. The plant can be configured for one, two or three degrees of freedom. For this lab, we prefer the two degree of freedom configuration since we would expect a fourth-order transfer function model, and have the ability to demonstrate many general features of the Bode magnitude plot.

The system has the flexibility of varying spring constants, masses of the carts, and the location of a viscous damper. For this lab, the instructor configures the first cart to have 1.0kg of added mass, and the second cart to have 2.0kg added. The first two carts are connected with a stiff spring. The damper is not connected, but the students' model should account for some damping due to rolling resistance of the bearings under each cart. Experimental uncertainty, differences in actual system mass, stiffness and rolling resistance will cause students' answers to vary from team to team.

The student obtains system frequency response data by following these steps:

1) Under the Command/Trajectory menu, set the open-loop input amplitude to 0.5 V with a frequency of 1 Hz. Set the number of repetitions equal to 10 times the frequency (in Hz).
2) Under Utility, select Zero Position
3) Select Command/Execute and press Run.

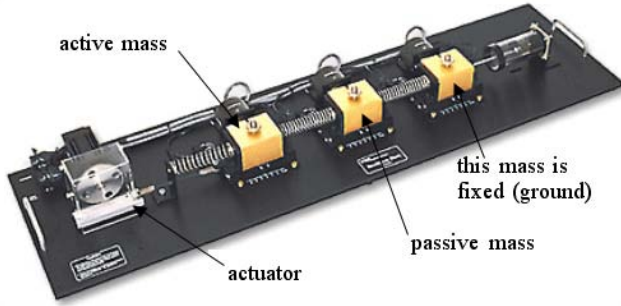Fig. 1.   The ECP Rectilinear Control System
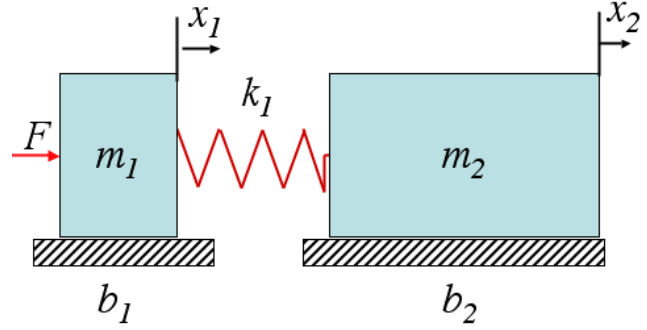

Fig. 2.   Schematic of Lumped Parameter Model

4) Watch the response. After the Upload successful dialog completes, click OK.
5) (Optional) To look at an individual data set, select Plotting/Setup Plot. Do this for some of the higher frequencies to insure that the system has reached steady state.
6) Export the data using the Data/Export Raw Data menu.
7) Repeat for the remaining frequencies of interest. As a minimum, use the frequencies [1 2 3 4 5 6 7 8 9 10] Hz.

IV. DATA REDUCTION

The preferred system configuration for this lab is shown in schematic form in Figure 2. The motor applies a force, F, directly to the first mass. The first two masses are connected by a single spring. The first cart, m_1, has 1.0kg of added mass. The second cart, m_2, has 2.0kg added. The first two carts are connected by a stiff spring, k_1. The damper is not connected, however the model must account for rolling resistance in the bearings under each cart–b_1 and b_2 respectively.

Applying first principles, and choosing the state vector $\begin{bmatrix} x_1 & \dot{x}_1 & x_2 & \dot{x}_2 \end{bmatrix}^T$, the resulting state space model is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}F \qquad (1)$$

where,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_1}{m_1} & -\frac{b_1}{m_1} & \frac{k_1}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_1}{m_2} & 0 & -\frac{k_1}{m_2} & -\frac{b_2}{m_2} \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} 0 & \frac{K}{m_1} & 0 & 0 \end{bmatrix}^T.$$

The constant K is included to account for the unknown static gain of the system. The position of each mass is measured via optical encoders. Thus the output equation is

$$\mathbf{y} = \mathbf{C}\mathbf{x} + DF \qquad (2)$$

where

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and $\mathbf{D} = 0$. The two system transfer functions are then found from the state space description. From F to x_1 the result is

$$\frac{X_1(s)}{F(s)} = \frac{K\left(m_2 s^2 + b_2 s + k_1\right)}{\Delta(s)}. \qquad (3)$$

From F to x_2 it is

$$\frac{X_2(s)}{F(s)} = \frac{Kk_1}{\Delta(s)}. \qquad (4)$$

The system characteristic equation is

$$\begin{aligned} \Delta(s) &= s\left(m_1 m_2 s^3 + \left(b_1 m_2 + b_2 m_1\right) s^2 \right. \\ &\quad + \left. \left(k_1\left(m_1 + m_2\right) + b_1 b_2\right) s + k_1\left(b_1 + b_2\right)\right). \end{aligned}$$

It should be evident from the two transfer functions that this system will illustrate several important features of typical system frequency response. The following features are clearly seen in the sample frequency response plot shown in Figure 3. The system characteristic equation has an origin pole due to the rigid body mode. This is shown by the 20dB/decade slope at low frequency. The three remaining poles can be deconvolved into a lightly damped second order pair due to the oscillatory mode, and seen as the 'peak' near 30 r/s, and one real pole. The first transfer function has a lightly damped second order zero pair which is seen as the deep 'notch' near 18 r/s.

If the mass of the first cart m_1 is known from previous experiments, it is possible to identify the remaining parameters by comparing the experimental frequency response data with a theoretical model, and using non-linear programming, search for the parametric values which minimize the sum square error between experimental and theoretical response magnitudes (in dB). The non-linear programming is facilitated by the Matlab function fminsearch which uses the Nelder-Mead simplex method. Convergence of this search is highly sensitive to the initial guess. The user must force the search to use magnitudes of the damping values, b_1, and b_2, and mass of the second cart m_2, in order to render a stable model, and accurately identify the parameters. In our lab sequence, the students have previously identified the mass of the first cart m_1 from the step response of a single
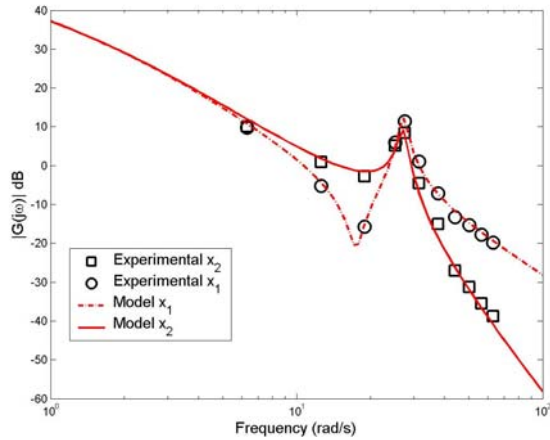
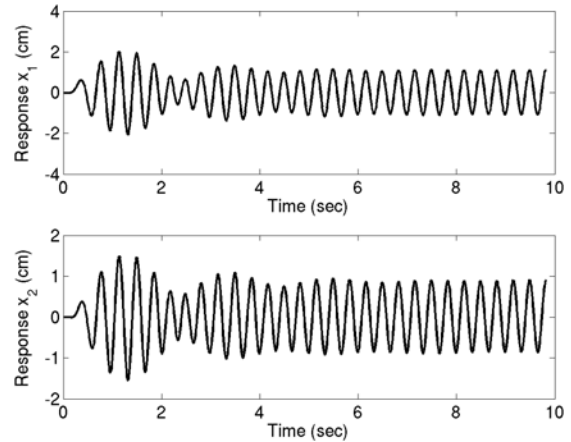Fig. 3.   Example Experimental Frequency Response with Theoretical Fit.

Fig. 4.   Typical System Sinusoidal Response

degree of freedom setup, therefore, they will have enough information to find a parametric model of the two degree of freedom system.

Oftentimes, the student will find it much easier to find a non-parametric model to fit the data first. That is, instead of finding m_2, k_1, b_i, and K, we will match the following transfer function.

$$\frac{X_2(s)}{F(s)} = \frac{K|p_2|\omega_{n3}}{s(s-p_2)(s^2 + 2\zeta_3\omega_{n3}s + \omega_{n3}^2)} \qquad (5)$$

Where the unknown values are K, ζ_3, ω_{n3}, and p_2. The three models above (state space, parametric transfer functions, and non-parametric transfer function) give the instructor a wide range of options in determining the level of rigor required from the students. In fact, when seeking a parametric state space model, it is often easiest to first identify the unknowns in the non-parametric model, then the parametric transfer functions, and finally the state space model. For instance, with poor initial guesses for natural frequency and damping ratio, the initial search may yield a good estimate of static gain, K, only. This estimate of K may then be used in a subsequent search to get convergence on the correct natural frequency and damping ratio, and so on.

Looking at a typical response such as Figure 4, the student should immediately recognize this as a linear system response to a sinusoidal input. That is, frequency is preserved in the output, and output can be related to input by a gain factor and phase lag. The input, which was omitted from the plot to reduce clutter, is a smooth sinusoid from the internal signal generator.

$$u(t) = A\sin\omega t \qquad (6)$$

$$\omega = 2\pi f \qquad (7)$$

In the case shown the input has an amplitude of 0.5 volts, and a frequency of 3Hz. The outputs show some transient behavior for the first six seconds of motion. In steady state

the response can be described by Eq. 8. In the case shown, the masses are swinging in phase with each other and approximately 180 degrees out of phase with the input.

$$x_i(t) = |G(j\omega)| A\sin[\omega t + \arg G(j\omega)], \qquad (8)$$

where $|G(j\omega)|$ is the transfer function magnitude. We subtracted the mean from input and output signals prior to plotting them on Figure 4. The student should do likewise prior to comparing maximum or minimum amplitudes in order to determine $|G(j\omega)|$. We expect the student to exercise some judgment in determining whether to use the maximum or minimum of the output oscillation in calculating the transfer function magnitude.

Figure 5 shows the ASCII format in which the ECP executive program saves the data when using Export Raw Data in step 6 above. Note that the open square bracket used in Matlab array assignments is placed at the beginning of the data, and data is arranged in columns using space delimiters, and semi-colon delimiters for rows. There is also a close square bracket at the end of the file which is not shown in the Figure.

Figure 6 shows the file after minimal editing by the student, changing the header to a comment, and the open square bracket to the array assignment"dat".

The Bode magnitude for each frequency can be determined from the ASCII data files using the following steps.

Extract the Encoder 2 Position data and subtract the mean.

Convert Encoder 2 Position to centimeters. The encoder sensitivity is 2196 counts/cm.

Now determine the ratio of output to input amplitude. Plot the data to determine where the oscillations have reached steady state. Output amplitude can then be determined by taking the maximum value from several steady state oscillations. Since the input amplitude was specified to be 0.5 volts, we divide by 0.5 to get the ratio of output magnitude over input magnitude.
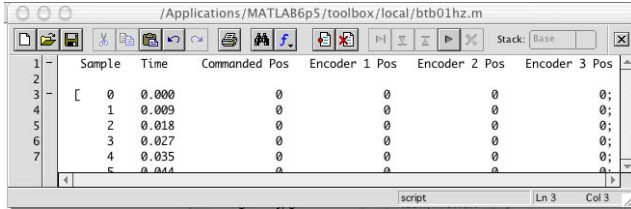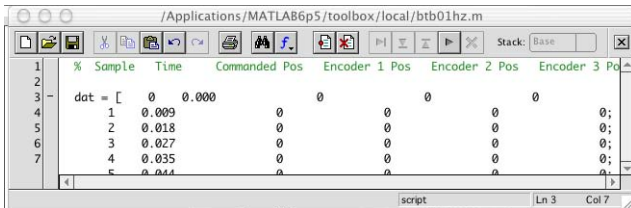
Fig. 5.   ECP Data Export Format

Fig. 6.   Data File After Editing

Note, the array indices in this statement need to be adjusted so that you consider only oscillations after the system has reached steady state. It is probably easiest to do these calculations inside each data file.

The student should then write a top-level script that 1) executes each data file. 2) Converts the magnitude rations to decibels and plots the experimental data on a semilog plot. 3) Uses fminsearch to find the model best fits the data. The instructor may require either a single non-parametric transfer function of Eq. 5, the set of parametric transfer functions of Eqs. 3 and 4, or the state space model of Eqs 1 and 2. The built-in Matlab bode function can be used to compute the theoretical magnitude data.

Fminsearch uses the Nelder-Mead Simplex approach which finds a local minima near the initial guess and does not require gradient information. The student should be reminded that his initial guess of the transfer function parameters can greatly influence the success of the search. When using the non-parametric models, the student can help insure convergence by fielding an initial guess for the natural frequency based on where the data apparently peaks. It would also be possible to guess the frequency of the zero in Eq. 3 by plotting the experimental response magnitude from F to x_1. The authors have found it best to first determine the system static gain K by optimizing a non-paramtric model. Then optimize the parameters of the parametric model of Eq 4, comparing only expermental response magnitude from F to x_2. Finally, optimize the parameters of the state space model, taking into account the response magnitude for both x_1 and x_2. (A Matlab script for performing this optimization problem is given in the Appendix.)

Plot the magnitude of your best fit transfer function, and determine the resonant frequency, and the zero frequency. The model can be greatly improved by then obtaining response data at these two frequencies. When operating near the system resonant frequency, you will probably need to

TABLE I
IDENTIFIED PARAMETERS OF THE SPRING MASS SYSTEM

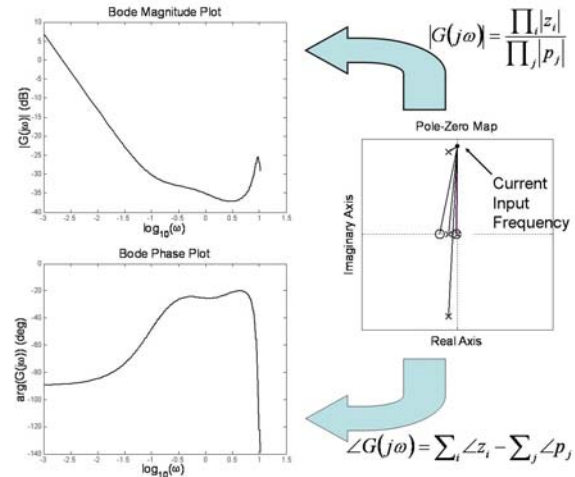| m_1	m_2	k_1	b_1	b_2	K
1.965	2.769	844.5	4.108	4.884	735.3

Fig. 7.   Graphical Evaluation of Frequency Response.

reduce the input amplitude to 0.25 volts to avoid exceeding the travel limits of the device. Incorporate this data into your analysis, and re-do the numerical fit analysis.

Figure 3 shows a typical Bode Magnitude plot with experimental data and the best theoretical fit transfer function response.

The best theoretical transfer functions for the example shown in Figure 3 are given by

$$\frac{X_1(s)}{F(s)} = \frac{374.2s^2 + 660.0s + 1.141(10)^5}{s\left(s^3 + 3.854s^2 + 738.5s + 1396\right)} \quad (9)$$

$$\frac{X_2(s)}{F(s)} = \frac{1.141(10)^5}{s\left(s^3 + 3.854s^2 + 738.5s + 1396\right)} \quad (10)$$

The parameters of these transfer function are given in Table 1.

From a transfer function like that shown in Eq. 5, the student may obtain poles that are clearly unstable, that is, in the right-hand s-plane. However, since we only considered the frequency response magnitudes, the identified poles could lie in either the left or right half plane. This is illustrated in Figure 7 which shows the relationship between the system pole-zero map and frequency response. An animated version of this figure is available from the authors. The frame on the right is the pole-zero map or s plane, where the X symbols represent system poles and the circles represent system zeros. The dot on the imaginary axis represents the current input frequency. The system Bode magnitude at this frequency is the ratio of product of lengths of line segments from zeros to excitation frequency over product of lengths

of line segments from poles to excitation frequency or:

$$|G(j\omega)| = \frac{\prod_i |z_i|}{\prod_j |p_j|} \quad (11)$$

The Bode magnitude plot is shown in the upper left frame. The animated version draws the magnitude and phase plots in sync with the pole-zero map animation. Thus, the magnitude and phase plots are truncated at the frequency corresponding to the current picture on the pole-zero map. Note that the two lightly damped poles could lie either in the left or right half plane, and this would not change the Bode magnitude of the system at all excitation frequencies. Thus, we would have the liberty to arbitrarily change the real part of the second pair of poles to be negative.

V. TEACHING CONSIDERATIONS

This lab provides the students an opportunity to see first hand what is meant by frequency response. Although the input signal is an electrical voltage, and not readily observable during data collection, using a two degree of freedom spring mass configuration introduces obvious phase differences between the controlled mass and the passive mass. This configuration also provides a transient response of significant length that the students can clearly observe, especially in frequencies near the resonant peaks.

Although the phase shift could be measured, doing so would add greatly to the data reduction effort. Also, ignoring the phase data provides an opportunity for the professor to emphasize the relationship between complex plane pole-zero map and system frequency response. Non-minimum phase implies that the significant difference between left and right half plane poles and zeros is that right half plane ones contribute significantly more phase lag or lead.

VI. CONCLUSIONS

In this paper, we have presented course background, lesson objectives, data collection and reduction techniques for a system identification laboratory. This laboratory is particularly well suited to introducing students to the physical meaning of frequency response. The data reduction is readily done in Matlab, since the ECP export format is easy to import into Matlab. The student can determine experimental frequency response magnitudes fairly easily but should plot each input/output data set to insure their calculations are based on steady-state behavior. The hardware we have chosen has imperfect sensors and this provides opportunity to address uncertainty in experimental data.

VII. ACKNOWLEDGMENT

REFERENCES

[1] Manual for Model 210/210a Rectilinear Control System, Educational Control Products, Bell Canyon, CA, 1999. http://www.ecpsystems.com
[2] MATLAB is Copyright The Mathworks, Inc. 24 Prime Park Way, Natick, MA 01760-1500. http://www.mathworks.com

VIII. APPENDIX

Listed below is an example Matlab script for performing some of the functions described in the paper.

```
% inside each data file:
dat1 = [ 0 0.000 ...
...4561 5147 0];
% Extract the steady-state position
% data and subtract the mean
Enc2cm = dat1(:,5) - mean(dat1(500:1000,5));
Enc2cm = Enc2cm/2196;
Enc1cm = dat1(:,4) - mean(dat1(500:1000,4));
Enc1cm = Enc1cm/2196;
command = dat1(:,3);
time = dat1(:,2);
Magy(2) = 2*max(Enc2cm(500:1000));
Magx(2) = 2*max(Enc1cm(500:1000));
% Run the optimization
x0(1) = 1.8862593; % guess for m2
x0(2) = 88.8674431; % guess for b1
x0(3) = 70.4184818; % guess for b2
x0(4) = 5.8338852e+002; % guess for k1
x0(5) = 1.5332170e+002; % guess for K
options = optimset(@fminsearch)
options = optimset(options,'Display','iter');
coeffs = fminsearch(@lab3,x0,options)
% Compute the sum squared error in dB
% between experimental and theoretical
% magnitudes.
function J = lab3(x)
m2 = abs(x(1));
m1 = 1.9649;
b1 = abs(x(2));
b2 = abs(x(3));
k1 = x(4);
K = x(5);
% enter state space matrices here
A = [0 1 0 0; -k1/m1, -b1/m1, k1/m1, 0;...
0 0 0 1; k1/m2, 0, -k1/m2, -b2/m2];
B = [0; 0; K/m1; 0];
C = [1 0 0 0; 0 0 1 0];
D = 0;
Gss = ss(A,B,C,D);
ww = 2*pi*[1:10];
maggie = bode(Gss,ww);
maggie = 20*log10(reshape(maggie,2,10));
magxdB = [ % put experimental data here %]';
magydB = [% put experimental data here %]';
J = norm(maggie(1,:)' - magxdB)...
+ norm(maggie(2,:)' - magydB);
```