Hands-on Learning with Computer Simulation Modules for Dynamic Systems

Hodge Jenkins[1]

Abstract – While the mathematical development and solution of dynamic equations of motion for mechanical systems is readily accomplished by engineering students, physical insights based on these equations is not. Interactions of multiple degrees of freedom as well as the consequences of modeling assumptions for dynamic systems are especially difficult for many students to grasp. This paper presents a series of in-class computer simulation modules on modeling of dynamic systems for hands-on learning, suitable for undergraduate courses in dynamic systems or vibrations. Simulations provide a means for students to visualize solutions and investigate alternative modeling or simplifications such as linearization of small angles, via graphical tools for plotting acceleration, velocity, and displacement results. Comparisons between linearized models and more complex nonlinear models help students determine the appropriate application of nonlinear modeling. These computer-based, visualization modules also offer students a vehicle for further inquiry, improved comprehension, and increased learning opportunities both inside and outside of class.

Simulation and modeling with Matlab® and Simulink®, of various mechanical systems was accomplished through four classroom modules. Module topics included: one degree of freedom (1-DOF) free vibration, 1-DOF forced vibration with harmonic forces, 2-DOF free and forced vibration, and simulation of an earthquake on a 4-DOF structure with actual acceleration data from the 1940 El Centro earthquake. Computer-generated solutions were compared to closed-form solutions for several of the studied systems, establishing a means for verification of results. Transfer function and state space representations of linear, time-independent, multiple DOF systems as well as graphical-based modeling of linear and nonlinear systems via a function block and signal flow diagrams were applied in the modules.

Instructional materials for the classroom computer exercises and the associated semester project, forming the basis of the modules, are provided and discussed in the paper. While resources for this module were developed for specific software, the concepts and topical problems presented for investigation may be used with other modeling and analysis software. The module materials may also be used by students as self-paced tutorials on modeling and simulation of dynamic systems.

Keywords: Vibration, modeling, simulation, computer-based learning

INTRODUCTION

Engineering undergraduate curriculums are changing to meet the needs and desires of modern engineering standards. Course topics are being integrated to reflect advancing knowledge, interests of students and faculty, and the available engineering tools. Many current engineering undergraduate programs allow greater flexibility in the course selection of undergraduate engineering [1-4]. In many mechanical engineering undergraduate curricula vibration has been relegated to an elective. System dynamics has assumed the role of providing the fundamentals of vibration in these cases. Some programs have combined system dynamics with control, further reducing the linear systems education opportunities. Many students fail to learn the basics of system modeling via this approach. In an

[1] Mercer University, Mechanical & Industrial Engineering Department, 1400 Coleman Ave., Macon, GA 31207, jenkins_he@mercer.edu

effort to partially address this educational concern a sequence of computer modules were developed to provide hands-on learning of discrete, lumped parameter systems with modern engineering simulation tools, such as Matlab and Simulink.

Matlab and Simulink have been used as mathematical tools by many engineering disciplines, especially in areas of dynamics systems, signal processing, and control. Most undergraduate students will encounter these software packages in courses related to dynamics systems, control, or numerical methods.

Computer simulations of mechanical systems are beneficial from an educational perspective. Previous studies indicated that instructional content as well as delivery format have substantial impact on student learning. Visualization has been shown to be an effective means for aiding student comprehension. As engineering students tend to be more visual and sequential learners, visualization is an important means to engage students in active learning experiences. [5]

Current mechanical engineering education literature includes many reports of novel efforts to increase student learning by supplementing traditional classroom activities with various forms of multimedia and alternative technology-based instruction. A recent study [6] has shown that computer-based instructional technology resulted in significantly higher student performance than traditional lecture formats. Conclusions, based on those results, attributed the improvement to increases in Time on Task, Student Interest, and Instructor Interest. In another study, preliminary results comparing the effectiveness of traditional lecture versus a computer-based finite element analysis tutor in a junior level mechanical engineering course showed that the ability of students given the computer-based instruction was 30% better than the students who received traditional instruction [7]. In that study the primary purpose of a computer-based module was to provide an experience equivalent (or better) to in-person delivery. It is clear from educational research that students learn in variety of modes: listening, watching, and doing.

Bloom's taxonomy defines six cognitive levels of learning. The lowest through highest levels are knowledge, comprehension, application, analysis, synthesis, and evaluation. [9] The modules and the associated semester project described in this paper try to target all six levels. It is generally recognized that hands-on learning (or interactive) learning is best for students to achieve higher levels of learning [10], many upper-level courses do not have such a component, outside of homework or a project. Several authors have hypothesized about the benefits of including Matlab or Simulink as part of a vibrations or systems dynamics course. [7, 8]

Although it is the desire of engineering professors to have students be able to synthesize and evaluation new analytic solutions, many engineering undergraduate students are reluctant to solve for symbolic mathematic relationships from engineering models. Instead of using variables to develop underlying relationships in a design, students prefer to place numbers in to equations as soon as possible to attain a single solution, missing observations of variable effects and obtaining general conclusions. While a significant number of undergraduate students eventually learn to use solutions in symbolic variable format, many do not. To overcome the student aversion of symbolic answers, simulation and plotting of equation results has been suggested. [8]

This paper reviews the content of classroom, computer-based simulation modules, created for vibration and system dynamics students at the senior level. The modules provide a hands-on learning experience for students.

MODULES

Outlined in this section are the four simulation modules for a senior mechanical engineering technical elective in vibration. Module topics: 1-DOF free vibration, 1-DOF forced vibration with harmonic forces, 2-DOF free and forced vibration, and simulation of an earthquake on a 4-DOF structure with actual acceleration data from the 1940 El Centro earthquake. Material given in the modules was intended to supplement that of the classroom lectures.

Two ways of performing simulations were demonstrated in the modules. Matlab was applied to linear models using transfer functions and state space matrix formulations. Simulink was demonstrated using nonlinear as well as linear models. Simulink highlights a graphical approach to modeling differential equations of motion as signal flow block diagrams. Matlab is text-based, with powerful matrix functions that may be combined in a program text file. Both methods are suitable for solution of the ordinary differential equations (ODE) involved in system dynamics. For most dynamis systems, linear ODEs are developed from linear time independent (LTI) parameters. Matlab is better suited to handle systems with larger numbers of degrees of freedom (DOF), while Simulink is more easily applied to non-linear equations and is preferred by new users for its graphical interface.

Goals for the modules are increased student interest in system dynamics, vibration, and analysis; increased student awareness of mechanical modeling implications; and improved student readiness for engineering practice.

Instructional materials developed for the laboratory sessions and a related project are discussed and provided in the appendix. Several computer-generated solutions are compared with simplified, closed-form solutions to provide bounding cases for verification of results. While materials for this module were developed for specific software, the concepts and topical problems presented for investigation may be used with other modeling and simulation software. The materials may also be used as a self-paced tutorial.

Module 1

Module 1 included an introduction or refresher on MATLAB and Simulink. Matlab was used with a text script of commands, while Simulink was used as graphical representation of dynamic equations, especially nonlinear equations.

Familiarization with the software was accomplished via a problem-based learning approach through modeling both of the one degree of freedom systems presented. A mass-spring-damper system and a damped pendulum under free vibration were modeled using the familiar application of Newton's second law of motion, Eq. (1) and Eq. (2). Simulations of linear time independent (LTI) systems are easily accomplished in Matlab using built-in functions for general solutions for systems with multiple forces and multiple degrees of freedom with initial conditions (e.g., lsim, step, ramp). Figure 1 shows the two single degree of freedom systems simulated in this first module. The numerical displacement solution of the mass-spring-damper system from initial conditions is easily compared to analytic solution, Eq. (3). When numerical and analytical solutions were plotted it together the curves are identical, visually demonstrating the validity of the numerical solution.

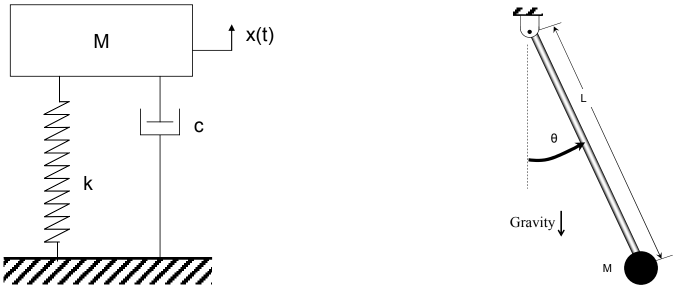

Figure 1. (a) Mass-spring-damper (b) Pendulum
Single degree of freedom systems modeled in Matlab and Simulink

$$m\ddot{x} + c\dot{x} + kx = 0 \text{ with } x(t=0) = x_o \text{ and } \dot{x}(t=0) = \dot{x}_o \tag{1}$$

$$ml^2\ddot{\theta} + mgl\sin\theta + c\dot{\theta} = 0 \text{ with } \theta(t=0) = \theta_o \text{ and } \dot{\theta}(t=0) = \dot{\theta}_o \tag{2}$$

$$x(t) = e^{-\omega_n t}\left\{x_o\cos\left(\sqrt{1-\zeta^2}\,\omega_n t\right) + \frac{\dot{x}_o + \zeta\omega_n x_o}{\sqrt{1-\zeta^2}\,\omega_n}\sin\left(\sqrt{1-\zeta^2}\,\omega_n t\right)\right\} \tag{3}$$

where $\omega_n = \sqrt{\dfrac{k}{m}}$ and $\zeta = \dfrac{c}{2m\omega_n}$

Consequences of modeling assumptions or parameter limitations are not always apparent to students, such as the small angle approximation (Eq. (4)) used in pendulum motion, depicted in Figure 1(b). The small angle assumption allows the linearization of the dynamic model, Eq. (2), to be reduced to Eq. (5). The closed-form analytic solutions of Eq. (1) and Eq. (5) are readily available and are similar in form to Eq. (3).

$$\sin\theta \approx \theta \tag{4}$$

$$\ddot{\theta} + \frac{c}{ml^2}\dot{\theta} + \frac{g}{l}\theta = 0 \tag{5}$$

Closed form solutions of the nonlinear dynamic equations without the small angle constraints are not easily generated. However, modeling of the nonlinear system in Simulink is easily accomplished, as seen in Figure 2. The Matlab linearized model developed from the Laplace transform and resulting transfer function, Eq. (5) is provided in the Appendix.

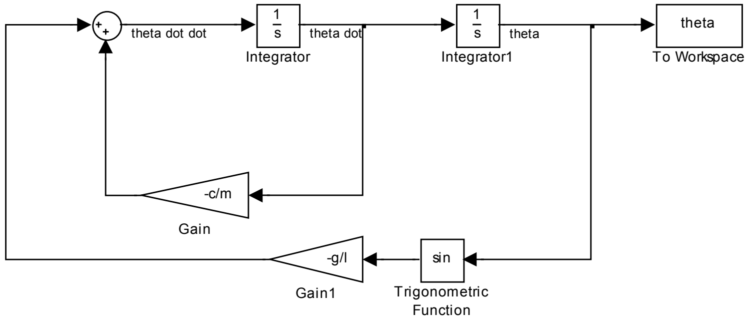

Figure 2. Nonlinear Simulink model for the single degree of freedom pendulum of Fig. 1(b)

Simulations with small and large angles (10˚ and 80˚) were run in Matlab and Simulink for comparisons. Figure 3 depicts the angular displacements of the pendulum for both the linear model (Eq. (5)) and non-linear model (Eq. (2)). As seen in Fig. 3(a) for a small initial angle displacement, both models yield the same result. However, the large initial angle highlights the nonlinearity of Eq. (2) as seen by the two distinct curves for the two models in Fig. 3(b).

Students then change the damping values to have over damped and critically damped motion simulations. The plot of the displacement easily demonstrates the differences in the system behavior. It should be noted that prior to the modules that students should be familiar with the Laplace transform and transfer functions. Also an introduction to state space is also suggested, to ease modeling of non-zero initial conditions. (Note: the initial condition simulation function in Matlab is only for use with state space models.)Homework was assigned to model and plot the position and velocity of a one degree of freedom system for varying initial conditions.

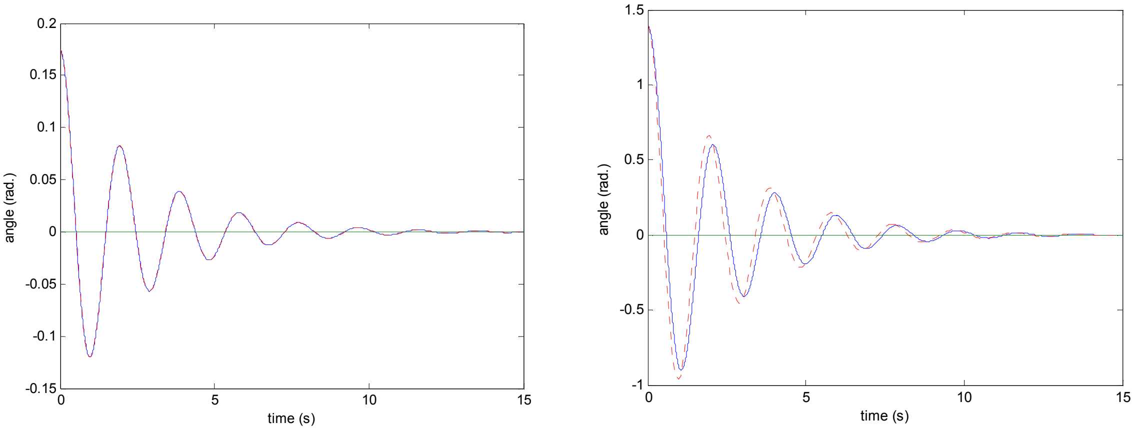

Figure 3. Angular position of pendulum vs. time for the linearized model (solid line) and non-linear model (dashed line) with initial conditions. The figure on the left is for an initial displacement of 0.175 radians (10˚) and the figure on the right started at 1.94 radians (80˚) of initial displacement.

Module 2

The second module explores the modeling of harmonic forces and the resulting motion for single degree of freedom systems. Students develop a comparison between viscous damping forces and Coulomb damping forces. In modeling Coulomb damping the $c\dot{x}$ term of Eq. (2) is replaced with a constant frictional force, $f_{friction}$, whose direction depends upon the direction velocity. The resulting dynamic equation is given in Eq. (6). A comparison of the viscous and Coulomb forces are plotted in Fig. 4 for sinusoidal motion. The Coulomb force is seen as a square wave signal, while the viscous force is sinusoidal.

$$m\ddot{x} + f_{friction}(sign(\dot{x})) + kx = 0 \qquad (6)$$

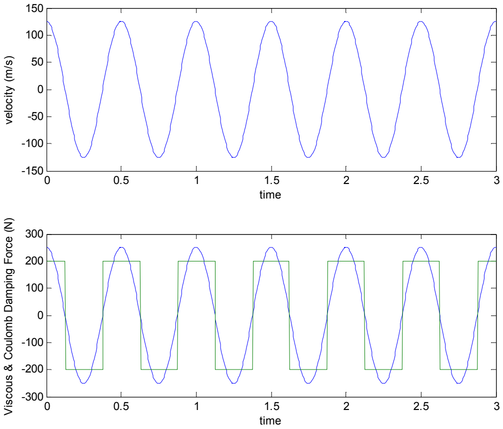

Figure 4. Top plot is velocity vs. time for sinusoidal motion. The bottom curve plots viscous and Coulomb damping.

The second part of this module develops harmonic forces and simulates the response of the mass-spring-damper system of Fig. 1(a) subject to those forces. The students are able to visualize the total solution, x_{total}, for the motion: the transient (homogeneous) and the steady state (particular). Students create three forcing functions, a single harmonic force, a second harmonic force of a different frequency, and the summation of the two together. All forces are plotted in Fig. 5 along with the displacement results. The numerical results are based on Eq. (7).

$$m\ddot{x} + c\dot{x} + kx = f(t) \text{ with } x(t=0) = x_o \text{ and } \dot{x}(t=0) = \dot{x}_o \qquad (7)$$

With
$$f(t) = F_1 \sin\omega_1 t + F_2 \sin\omega_2 t$$

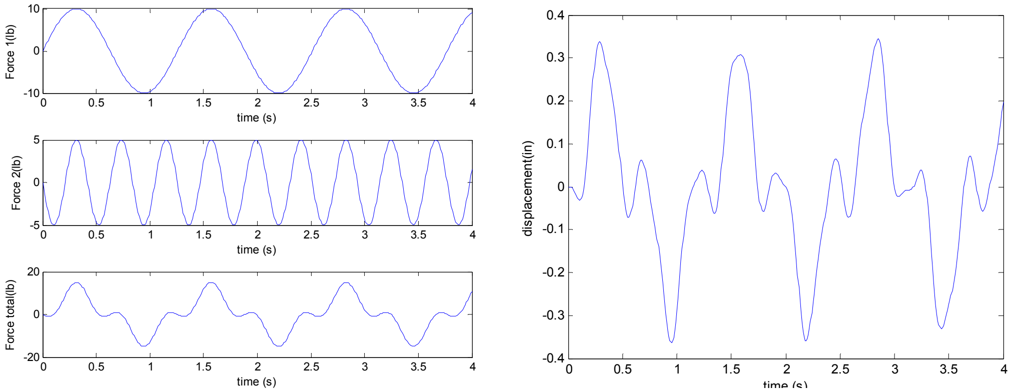

Figure 5. The total force and its two sinusoidal components are plotted (left). The resulting displacement (right) of the 1-DOF system is plotted for the total applied force.

Module 3

The third module continues with the modeling of two degree of freedom systems for both free and forced responses. Figure 6 depicts the modeled 2-DOF, mass-spring-damper system. Both Matlab and Simulink models were developed for comparison of methods and results. Figure 7 depicts the Simulink model, while the Matlab model is listed in the Appendix.

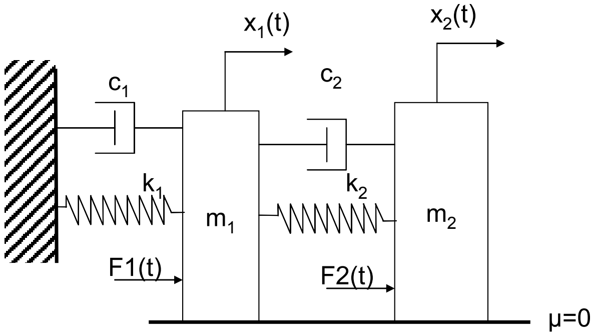

Figure 6. Two degree of freedom mass-spring-damper system.

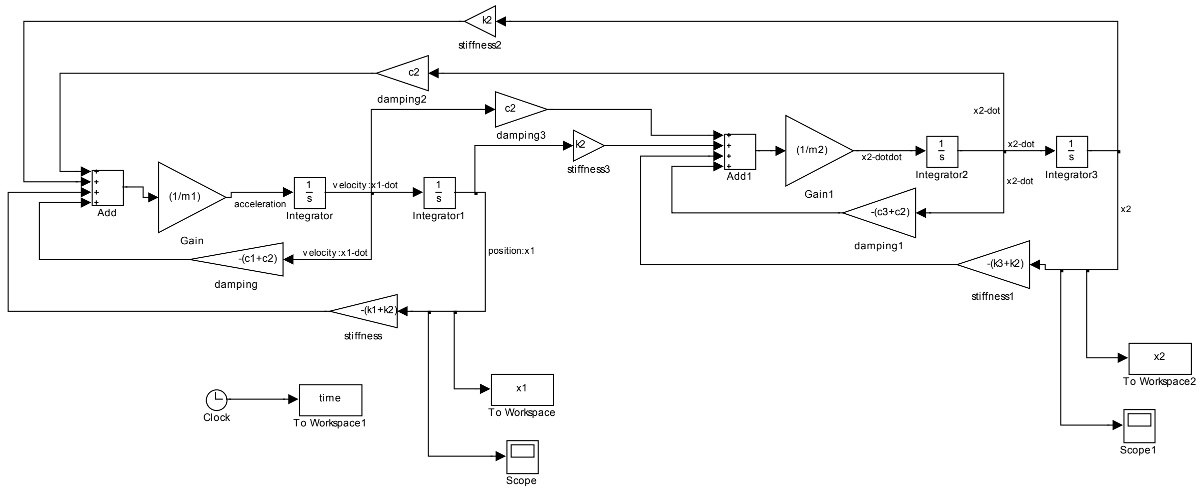

Figure 7. Simulink model for the two degree of freedom mass-spring-damper system.

Clearly as seen from Fig. 7, as the number of DOF increase the text-based Matlab model is easier to handle the modeling. For systems with more than one DOF, state-space representation is the most efficient way to represent the system in Matlab, Eq. (8), (9), (10) and (11). The model response to initial displacements under free vibration and to a combined initial displacements and sinusoidal forcing functions were simulated. The resulting displacements, with initial displacements, are plotted in Fig. 8. As observed in the figure, the transient response dies off relatively rapidly for both masses.

$$\{\dot{x}\} = [A]\{x\} + [B]\{f\} \tag{8}$$

$$\{y\} = [C]\{x\} + [D]\{f\} \tag{9}$$

Where $\{x\} = \begin{bmatrix} X_1 & X_2 & \dot{X}_1 & \dot{X}_2 \end{bmatrix}^T$ and $\{f\} = \begin{bmatrix} F_1 & F_2 \end{bmatrix}^T$

$$F_1(t) = 20 \sin(\omega_{f1} t)\ \textit{lbs.} \tag{10}$$

$$F_2(t) = 10 \sin(\omega_{f2} t + \pi)\ \textit{lbs.} \tag{11}$$

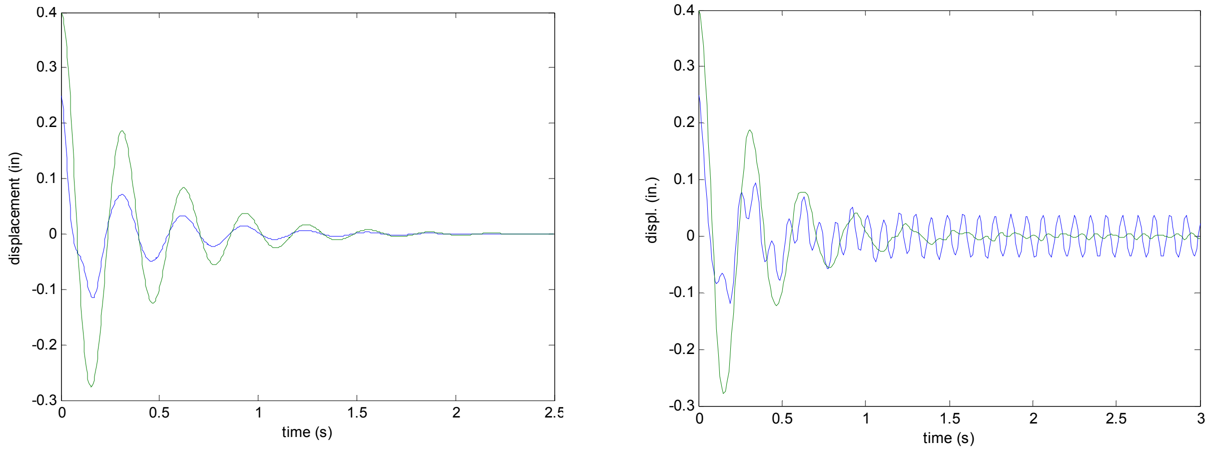

Figure 8. **(a) Free vibration with initial conditions** **(b) Forced vibration with initial conditions**
Displacement results for the two masses of Fig. 6.

Module 4
The final module is really an extension of the third. The state space representation is expanded to 4 DOF, as seen by the system given in Fig. 9. Using the state-space representation of the system, the natural frequencies are easily found using the eigen value function in Matlab. With the standard eigen function output provided, mode shapes can be plotted.

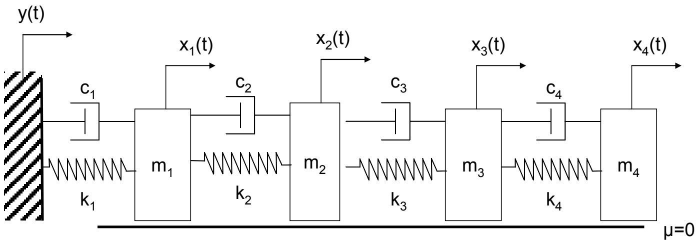

Figure 9. Model for the four degree of freedom mass-spring-damper system with ground motion.

To make the problem more interesting for students, real earthquake data was used as input to the system. The N-S acceleration history 1940 El Centro earthquake[13] was used as the source of ground motion. To obtain the ground forces (Eq. (12)) applied to the first mass (m_1), ground displacement and velocity are required. These inputs are obtained by numerically integrating the acceleration date. The results of the simulation are the displacements of each mass and the applied ground force of Eq. (12). These are plotted in Fig. 11. The Matlab model is included in the Appendix.

$$F_{ground} = c_1\dot{y} + k_1 y \tag{12}$$

Course Project

A course simulation project was created to have students build on the four modules and demonstrate their understanding of vibration analysis and Matlab through a comprehensive design project. The project was to design and analyze a cell tower structure for vibration from harmonic wind loads as well as an earthquake. The description of the project is included in Ref. [14].

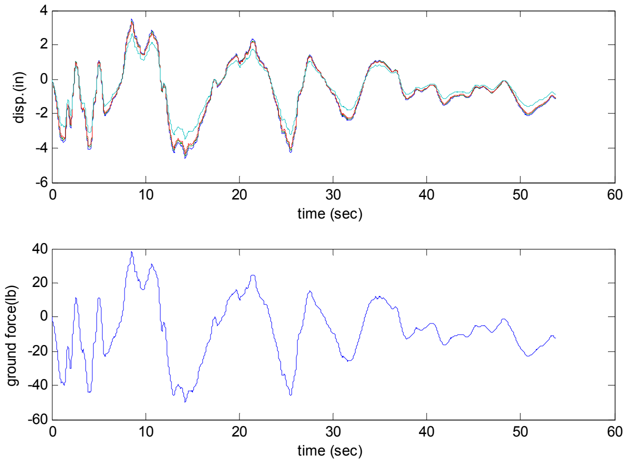

Figure 11. Displacements of each of the four masses and the applied ground force of the 1940 North-South El Centro earthquake simulation.

SUMMARY

The four modules presented here formed the basis of a computer-based simulation component of a vibration course. Material given in the modules was intended to supplement that of the lecture. In effect, the modules might be viewed as a laboratory, albeit a virtual one. The benefits attained from these hands-on modules are multifold. The simulation and modeling of the vibration systems in the modules stimulated student interest in the course. Students sharpened their skills in interpretation of results through plotting functions and simulation data. Most students greatly increased their abilities in applications of Matlab and Simulink.

A shortcoming of the modules was that they were started later in the semester than optimal (about 5 weeks into the 15-week semester). While students were familiar with Matlab and a few had seen Simulink, most of the students were not comfortable in using the software at the beginning of the course. An additional module on general Matlab and Simulink use at the beginning of the term with associated computer-based assignments would have been helpful for the students.

While the mathematical models and simulations presented here were helpful in student learning, additional modules should be developed. For a different perspective on vibration, a finite element approach may have some merit when continuous systems are discussed.

REFERENCES

[1] URL: http://meche.mit.edu/academic/undergraduate/course2a/, November 24, 2007.

[2] URL: http://www.mechse.uiuc.edu/media/pdfs/for/undergraduates/me/me_curriculum.pdf, November 24, 2007.

[3] URL: http://ughb.stanford.edu/OSA/handbook/handbookfiles/handbooks/07-08/5-ProgramDescriptionsAndRequirementsForEngineeringMajors.pdf, November 24, 2007.

[4] URL: http://www.coe.berkeley.edu/current_students/announce/eng-announce-07-08.pdf , November 24, 2007.

[5] Kolari, S. and Savander-Ranne, C., "Visualization Promotes Apprehension and Comprehension," *Int. J. Eng. Educ.*, Vol. 20, No. 3, 2004,pp. 484-493.

[6] Rutz, E., R. Eckart, J. Wade, C. Maltbie, C. Rafter, and V. Elkins, "Student Performance and Acceptance of Instructional Technology: Comparing Technology-Enhanced and Traditional Instruction for a Course in Statics," *J. of Eng. Educ.*, Vol. 92, No. 2, 2003, pp. 133-140.

[7] Milton-Benoit, J. Grosse, I.R. Poli, C. and Woolf, B.P., "The Multimedia Finite Element Modeling and Analysis Tutor," *J. of Eng. Educ.* Vol. 87, 1998, pp. 511-517.

[8] Cai, L.W., “Using Matlab to Enhance Engineering Students’ Analytical Problem-solving Skills,” IMECE2007-43020, 2007, pp. 1-10.

[9] Carlson, L. E. and J.F. Sullivan, “Hands-on Engineering: Learning by Doing in the Integrated Teaching and Learning Program,” *Int. J. Eng. Educ*., Vol. 15, No. 1, 1999, pp. 20-31.

[10] Regan, M. and S. Sheppard, "Interactive Multimedia Courseware and the Hands-on Learning Experience: An Assessment Study", *J. of Eng. Educ.* Vol. 85, 1996, pages 123-131.

[11] Campbell, J.O., J.R. Bourne., P.J. Mosterman, and A.J. Brodersen, “The Effectiveness of Learning Simulations for Electronic Laboratories,” *Int. J. Eng. Educ.*, Vol. 18, No. 1, 2002, pp. 81-87.

[12] Sabu, John, Dean Reeves, Robert Evans and Ben Bundy, “Smart systems used as an educational aid in mechanical engineering,” *World Transactions on Engineering and Technology Education*, Vol.3, No.1, 2004.

[13] URL:http://www.vibrationdata.com/elcentro_NS.dat , November 24, 2007

[14] URL: http://faculty.mercer.edu/jenkins_he/vibration.htm, November 24, 2007.

Hodge E. Jenkins

Dr. Hodge Jenkins is an Assistant Professor of Mechanical Engineering in the Department of Mechanical and Industrial Engineering at Mercer University in Macon, Georgia. Prior to coming to Mercer in 2002, Dr. Jenkins was employed in optical fiber product development with Bell Laboratories of Lucent Technologies. He is a registered professional engineer, and with over 20 years of design and development experience in high-precision machine design, dynamic analysis, process automation, control, and robotics. Dr. Jenkins holds a Ph.D. in Mechanical Engineering from Georgia Institute of Technology in (1996), as well as BSME (1981) and MSME (1985) degrees from the University of Pittsburgh. His professional affiliations include ASEE and ASME.

Appendix

Module 1 Models

```
%% linearized pendulum

g=386.4 %% in/s2
m=10/g %%lb-s2/in
c=0.02 %% lb-s/in
l=36.0 %%% inches
A=[0 1
   -g/l -c/m]
B=[0;0]
C=[1 0; 0 1]
D=[0;0]
xo=[10/180*pi 0]
sys=tf(A,B,C,D)
[Y,T,X]=initial(sys,xo)
plot(T,Y(:,1))
xlabel('time (s)');
ylabel('angle (rad.)');
```

```
%% mass spring damper
%% free vibration
w=100. % weight lb
g=386.4 % in/s2
k=50. % lb/in
zi=0.1 %% damping ratio
%% initial conditions
xo=0.25 % inches
xdoto=0  % in/s

m=w/g
wn=sqrt(k/m)
c=zi*2*m*wn
NUM=[1]
DEN=[m c k]
[A,B,C,D] = tf2ss(NUM,DEN)
B=[0;0]
sys = ss(A,B,C,D)
ic=[xdoto  xo ]'
[y,t,x] = initial(sys,ic)
plot(t,y)
```

Module 2 Model

```
% 1 DOF, FORCED VIB
%WITH DAMPING
w=20. % weight lb
g=386.4 % in/s2
k=50. % lb/in
zi=0.0 %% damping ratio
%% initial conditions
xo=0.0 % inches
xdoto=0  % in/s
wf1=5;
wf2=15
t=(0:.01:4)';
F1=10*sin(wf1*t)
F2=5*sin(wf2*t+pi)
Ft=F1+F2

subplot(3,1,1), plot(t,F1)
xlabel('time (s)'), ylabel('Force 1 (lb)')
subplot(3,1,2), plot(t,F2)
xlabel('time (s)'), ylabel('Force 2 (lb)')
subplot(3,1,3), plot(t,Ft)
xlabel('time (s)'), ylabel('Force
total(lb)')

m=w/g
wn=sqrt(k/m)
c=zi*2*m*wn
NUM=[1]
DEN=[m c k]
Sys = tf(NUM,DEN)
ic=[xdoto  xo ]'
figure
[y,X]=lsim(A,B,C,D,Ft,t,ic)
plot(t,y)
xlabel('time (s)')
ylabel('displacement(in)')
```

Module 3 2 DOF simulation

```
% 2DOF, FORCED
%VIB WITH DAMPING
w1=100. % weight lb
w2=w1
g=386.4 % in/s2
k1=500. % lb/in
k2=k1/2
k3=k1/3
zi=0.1 %% damping ratio
%% initial conditions
x1o=0.25 % inches
x1doto=0  % in/s
x2o=0.40 % inches
x2doto=0  % in/s

m1=w1/g
wn1=(k1/m1)^(.5)
c1=zi*2*m1*wn1

m2=w2/g*3
wn2=(k2/m2)^(.5)
c2=zi*2*m2*wn2

c3=c2

A=[0 0 1 0
   0 0 0 1
    -(k1+k2)/m1  k2/m1 -(c1+c2)/m1 c2/m1
    k2/m2  -(k2+k3)/m2  c2/m2  -(c3+c2)/m2]
B=[0 0 1/m1 0; 0 0 0 1/m2]'
C=[1 0 0 0; 0 1 0 0 ]
D=[0 0 ; 0 0]'

sys = ss(A,B,C,D)
ic=[x1o  x2o   x1doto  x2doto]'

wf1=wn1*1.5;
wf2=wn1*2;
t=(0:.01:3)';
F1=20*sin(wf1*t)
F2=10*sin(wf2*t+pi)
Ft=[F1 F2]

subplot(2,1,1), plot(t,F1)
subplot(2,1,2), plot(t,F2)

figure
[y,X]=lsim(A,B,C,D,Ft,t,ic)
plot(t,y)
```

Module 4 Earthquake simulation 4DOF

```
g=386.4; %% in/s^2
m1=10/g; %lb/g
m2=10/g; %lb/g
m3=10/g; %lb/g
m4=10/g; %lb/g

k1=10; %% lb/in
k2=20; %% lb/in
k3=30; %% lb/in
k4=4; %% lb/in
k5=1; %% lb/in

c1=0.5; %% lb s/in
c2=0.5; %% lb s/in
c3=0.5; %% lb s/in
c4=0.5; %% lb s/in
c5=0.5; %% lb s/in

A=[-(k1+k2)/m1  k2/m1  0  0    -(c1+c2)/m1  c2/m1  0  0
  k2/m2  -(k2+k3)/m2  k3/m2 0  c2/m2 -(c2+c3)/m2 c3/m2 0
  0 k3/m3 -(k3+k4)/m3  k4/m3   0 c3/m3 -(c3+c4)/m3 c4/m3
    0  0  k4/m4  -(k4+k5)/m4    0  0  c4/m4  -(c4+c5)/m4
    0  0  0 0 1 0 0 0
    0  0  0 0 0 1 0 0
    0  0  0 0 0 0 1 0
    0  0  0 0 0 0 0 1]

B= [1/m1 0 0 0 0 0 0 0]'

C=[1 0 0 0 0 0 0 0
   0 1 0 0 0 0 0 0
   0 0 1 0 0 0 0 0
   0 0 0 1 0 0 0 0]

D=0

f1=c1*grd_vel +k1*grd_disp;

[Y,X]=lsim(A,B,C,D,f1,time)

subplot(2,1,1)
plot(time',Y)
xlabel('time (sec) ')
ylabel('disp.(in)')
subplot(2,1,2)
plot(time',f1)
xlabel('time (sec) ')
ylabel('ground force(lb)')
```