

# Enhancing Mechanical Engineering Deep Learning Approach by Integrating MATLAB/Simulink\*

MOSTAFA S. HABIB

Department of Mechanical Engineering, University of Bahrain, Bahrain. E-mail:  
mshabib@batelco.com.bh

*The methodology of integrating MATLAB/Simulink into mechanical engineering curricula to improve the quality of engineering education is presented and discussed. The integration is implemented, wherever appropriate, via MATLAB and Simulink. The four key elements in course design for deep learning approach, which include high motivation, active learning, peer interaction and well-structured knowledge base, are enhanced by using MATLAB and Simulink facilities. Among the benefits of integrating MATLAB and Simulink facilities are the emphasis on basic concepts, deep understanding, student's interactive learning, addressing real-world problems, graphical visualization, and the use of mathematical libraries. The methodology is illustrated through examples covering some mechanical engineering undergraduate courses which demonstrate significant improvement of the mechanical engineering deep learning approach.*

## INTRODUCTION

MANY RESEARCHERS have identified students' approach to learning as the key towards predicting student performance and the quality of learning outcomes [1, 2]. The student attitudes towards learning can be classified as:

- deep approach, when the student tries to make sense of what is to be learned;
- surface approach, when the student simply attempts to memorize facts.

From this classification it seems obvious that the quality of student learning can be improved by fostering a deep approach. Leading educationalists have pointed out four key elements in course design for a deep approach: *high motivation, active learning, peer interaction and a well-structured knowledge base* [3]. The role of instructor to implement different strategies towards promoting a deep approach to learning was emphasized in [4]. Goodlad [5] argues that the university teachers need to be familiar with the research on student learning so as to lead the students to the deep structures of ideas, rather than to generate the surface approach.

There are some practical problems with taking radical steps in the direction of encouraging the deep approach to learning [6]. Recent studies have shown that students' perceptions of teaching quality appear to depend upon their own beliefs about teaching and learning and those of their teacher [7]. Disturbingly, innovative non-traditional forms of teaching may not be appreciated and are rated

lowly by students with passive beliefs about learning. The studies also showed that a higher proportion of students had a passive conception of learning [7], and their views expressed in standard evaluation questionnaires could discourage innovation in teaching. The reason for such negative attitude of the students towards innovative teaching is related to the methodology of teaching. Many instructors who have introduced innovative way of teaching may not make use of the modern software technology and computer facilities such as MATLAB and Simulink in fostering the elements of deep learning approach. This requires that the instructors should spend more efforts to learn how to integrate these software technologies in their classes.

This work, therefore, presents the experience of the author in improving student learning by enhancing the elements of deep learning approach by making use of the advancement in the area of software technologies focusing on MATLAB and Simulink as an integral part of mechanical engineering course content.

## MAIN FEATURES OF MATLAB/SIMULINK

The integration of MATLAB/Simulink into mechanical engineering curricula is implemented wherever appropriate via two streams, MATLAB and Simulink, where the student can produce graphical animations that show the progress of a simulation visually. This significantly enhances the student's understanding of the system's behavior which achieves the main objectives of improving the quality of student learning. Based on the software

\* Accepted 29 June 2005.

characteristics of each stream and the engineering problems in hand, the instructor may select the appropriate stream to use in teaching.

### MATLAB

MATLAB is used in numerical simulation of engineering problems where mathematical models are available. The user (student) develops a code (program) to solve the mathematical model and may put the results in graphs where he/she can conclude the effect of various parameters and variables on a particular function by visualizations. The popularity of MATLAB is partly due to its long history, and thus it is well developed and well tested. Its popularity is also due to the following characteristics:

- It is significantly simpler in programming,
- Unlike FORTRAN, it does not need data types declaration,
- Programming compactness (user can do a lot with a code of few lines),
- Variables can support real, integer, string, and imaginary numbers (i. e.; continuity among integer, real and complex variables),
- It has a comprehensive mathematical library (ode45, eig, inv, . . .)
- It has extensive graphics tools including GUI functions,
- Animation capability (movie)
- Capability of linking with traditional programming languages,
- It is extensible (currently more than 25 ‘tool-boxes’ in various specializations) and
- Possibility of using Simulink from within MATLAB.

Therefore, in many schools a MATLAB course has replaced the traditional FORTRAN course and it is the principal computational tool used throughout the curriculum. In some technical specialties, such as signal processing, and control systems it is the standard software package for analysis and design.

### Simulink

Simulink is built on top of MATLAB. Mechanical engineering problems can be modeled using Simulink graphical libraries in the form of block diagrams. By using Simulink, writing the mathematical model in programming language is eliminated and the fact that the ‘code’ (program) which is the block diagram itself eliminates the risk that the program may not accurately implement the block diagram. Communication of Simulink with MATLAB is straightforward by exporting (importing) variables to (from) the workspace.

Simulink enables the user (student) to quickly create the model then analyze it to understand the performance and behavior. It is worth mentioning here that the student creates the model from the graphical libraries’ tools (Blocks) and can:

- explore the performance of his/her created model and refine to get better performance,

- analyze his model changes much faster;
- run analyses and explore what-if scenarios until he/she obtains the proper design.

In Simulink the student does not have to develop a program for the mathematical model of the system.

## METHODOLOGY

The methodology of integrating MATLAB software into mechanical engineering curricula to improve the quality of engineering education and enhance the deep learning approach is presented. The integration is implemented, wherever appropriate, via the two streams mentioned above where the student can produce graphical animations that show the progress of a simulation visually which significantly enhances his/her understanding of the system’s behavior. This meets the objectives of using MATLAB/Simulink in engineering education. The methodology is demonstrated via examples covering some undergraduate areas. The methodology used here includes:

- emphasize on basic concepts;
- deep understanding;
- student’s interactive learning;
- addressing real world problems;
- employing graphical visualization;
- use of mathematical libraries;
- redistribution of course contents.

### Emphasize on the basic concepts

It is important for the instructor to use basic concepts in explaining a topic wherever possible. This helps in building student’s systematic way of thinking. Thanks to the application software such as MATLAB/Simulink that helps the instructor to focus on the basic concepts. To demonstrate how an instructor can concentrate on the basic concepts in classroom and in the same time allows the student to be active in class; the following example serves in this direction.

**Example 1:** Determine the Laplace transform of the time delayed triangular pulse shown in Fig. 1.

**Solution:** The instructor together with the students may follow the following steps in solving this problem.

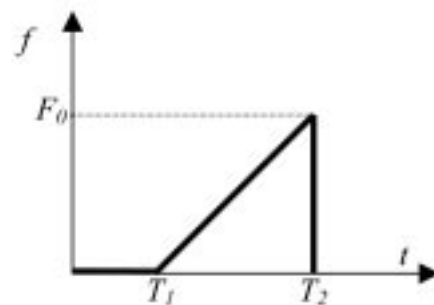


Fig. 1. The function given for Example 1.

*Step 1:* Write down the definition of Laplace transform of the given function  $f(t)$  as:

$$F(s) \equiv \int_0^{\infty} f(t)e^{-st} dt \quad (1)$$

*Step 2:* determine the formula of the given triangular pulse  $f(t)$  as:

$$\begin{aligned} f(t) &= 0 & \text{for } 0 \leq t \leq T_1 \\ f(t) &= \frac{F_0}{T_2 - T_1}(t - T_1) & \text{for } T_1 < t \leq T_2 \\ f(t) &= 0 & \text{for } t > T_2 \end{aligned} \quad (2)$$

*Step 3:* use the symbolic processor tool box in MATLAB to obtain the Laplace transform of  $f(t)$  given by Equation (2) by substituting in Equation (1). This can be determined by using the following MATLAB command:

Variables and constants

`syms F0 T1 T2 t s,`

Eqn. (1) and (2). Variable ↓

`Fs = int F0/(T2 - T1) * (t - T1) * rc[(-s * t, t`  
`limits`  
`T1, T2)`

$$Fs = -F_0 * (\exp(-s * T_2) * s * T_2 + \exp(-s * T_2) - \exp(-s * T_2) * T_1 * s - \exp(-T_1 * s)) / (T_2 - T_1) / (T_2 - T_1) s^2$$

*Step 4:* simplify the result:

`pretty(collect(Fs, exp(-s * T2)))`

$$F(s) \frac{F_0 \exp(-s T_2)(s T_2 + 1 - T_1 s) F_0 \exp(-T_1 s)}{(T_2 - T_1) s^2 (T_2 - T_1) s^2}$$

Or

$$F(s) = \frac{F_0}{(T_2 - T_1)s^2} (e^{(-sT_2)}(s(T_2 - T_1) + 1) + e^{(-sT_1)})$$

In this example the instructor together with the students applied the basic definition of Laplace transform which is possible to do fast during the class session. This enhances the systematic way of thinking, provides computer skills in the same time, and attracts the students' attention by being active.

#### Deep understanding

In teaching mechanical engineering courses such as dynamics of machines, vibration, and control, the instructor is encouraged to demonstrate the basic theory and give simple examples to make the students grasp the basic idea. He may use MATLAB/Simulink facilities to enhance the students' deep understanding. On the other hand, having understood the concept, the student can solve not only simple exercises but also hard ones. MATLAB enables the instructor to assign varieties

of problems ranging from simple to hard ones which reflect students' deep understanding. This approach is practiced by the author and can be demonstrated by the example shown in Appendix 1. This example represents a typical control problem which deals with the Root Locus (RL) method to design a proper compensator. The MATLAB control tool box provides various commands which help the student in plotting the RL of the system and enables the user to select (design) a proper compensator fast and accurate.

#### Student interactive learning

While explaining a topic, the instructor may give some examples to demonstrate the basic idea. It is highly recommended that he/she allows the students to solve the example in hand with him. This serves as:

- attracting the student's attention,
- gaining computer skills ( see Example 1);
- exposing the students' understanding to the explained topic (trouble shooting and fault diagnosis for the student's understanding).

#### Addressing real-world problems

The most important factor in engineering education is to allow the students to practice solving real-world problems as much as possible. This approach provides the students with the knowledge and experience to be able to face the practical engineering life after graduation and hence closing the gap between pre- and post-graduation practices. Engineering mathematical models are only simplified approximations of the real-world problems. MATLAB/Simulink allows accounting for real-world phenomena such as uncertainties, dead zone, backlash, saturation, etc . . . , which is expected to give reliable results (Appendix 1 and 2).

#### Graphical visualization

One important advantage of using MATLAB/Simulink in engineering curriculum is its capabilities of easily producing graphical visualization of the effect of the independent variables and parameters on the function combining them. This enhances deep understanding education approach by helping the student grasp the basic ideas fast and correct. It may include surface plots, 2D and 3D plots and movies, (see Examples a-1 and a-2).

#### Mathematical libraries

There are many engineering applications that require using mathematical libraries in computations (e.g, identities, integration, approximating functions, eigen-value problem, and solution of algebraic and differential equations, etc.). MATLAB/Simulink and their tool boxes provide such powerful scientific mathematical libraries (see examples 1, a-1, and a-2).

#### Rescheduling of course contents

In many engineering courses the time allocated to cover some topics can be reduced when the instructor decides to use computer facilities and

software programs throughout the curriculum. The semester's breakdown of most of the engineering courses' contents have still been following those where no such MATLAB facilities are used. Therefore, the instructor may redistribute the allocated time for the course contents making use of the computer facilities he/she planned to use. As an example, in the undergraduate control course the time allocated to manually sketch the root locus and frequency response diagrams to be used in compensator design is about 25–30% of the course time. The instructor can use MATLAB facilities instead to get the root locus, Bode, and/or Nyquist plots. He/she just has to explain the basic theory of how to obtain these plots, state the objectives of plotting them and then go ahead to use MATLAB to do the task. The time saved here may be used to address real-world and design problems. It is worth to mention here that those computer plots can be obtained easily, accurately and fast (see example a-1).

### CONCLUDING REMARKS

The methodology of integrating MATLAB/Simulink into mechanical engineering curricula to improve the quality of engineering education was presented and discussed. The integration was implemented, wherever appropriate, via MATLAB and

Simulink. The four key elements in course design for deep learning approach which include: *high motivation, active learning, peer interaction and well-structured knowledge base* were enhanced by using MATLAB and Simulink facilities. Among the benefits of integrating MATLAB and Simulink facilities were the emphasis on *basic concepts, deep understanding, student's interactive learning, addressing real world problems, employing graphical visualization, and the use of mathematical libraries*. The methodology was demonstrated via examples covering some mechanical engineering undergraduate courses which showed significant improvement of the quality of student learning.

The methodology of integrating MATLAB and Simulink into the engineering curricula is demonstrated via two examples. Example a-1 is presented in Appendix 1 and Example a-2 is given in Appendix 2. Example a-1 demonstrates the importance of integrating MATLAB into the contents of the undergraduate control course to solve a quasi-real-world problem. It asks for the design of an appropriate compensator where plant uncertainty exists. Such problems support the deep educational approach. In Example a-2, Simulink is used to model a second-order system with a harmonic forcing function with a dead zone. As explained earlier it was very easy to construct the model and get results very fast and accurate.

### REFERENCES

1. G. Gibbs, *Improving the Quality of Student Learning*, Bristol: Technical and Educational Services (1992).
2. G. Gibbs, Improving the quality of student learning through course design, in R. Barnett (ed.), *Learning to Effect*, Buckingham: The Society for Research into Higher Education and Open University Press (1992), pp. 149–165.
3. J. Biggs, Approaches to the enhancement of tertiary teaching, *Higher Education Research and Development*, 1989, pp. 7–25.
4. S. Toohey, *Designing Courses for Higher Education*, Buckingham: The Society for Research into Higher Education and Open University Press (1999).
5. S. Goodlad, *The Quest for Quality*, Buckingham: The Society for Research into Higher Education and Open University Press (1995).
6. A. A. Popov, Final undergraduate project in engineering: towards more efficient and effective tutorials, *Eur. J. Eng. Ed.*, **28**(1), 2003, pp. 17–27.
7. D. Kember and A. Wong, Implications for evaluation from a study of students' perceptions of good and poor teaching, *Higher Education*, **40**, 2000, pp. 69–97.
8. G. Franklin, J. Powel and A. Naeni, *Feedback Control of Dynamic Systems*, 3<sup>rd</sup> Edn, Addison Wesley.
9. W. J. Palm, *Introduction to MATLAB 7 for Engineers*, McGraw-Hill, 2005.

### APPENDIX 1

*Example a-1 (Bahrain University, MEG 482 Final Exam, June 2004)*

Consider the system shown in Fig. a-1.1; it is required to use root locus method to design a controller  $G_c(s)$  so that the closed-loop step response has a rise time  $t_r = 0.1$  s. and an overshoot of less than 12% [8].

**a-1.1** Verify that the transfer function relating the input force  $u(t)$  and the position  $y(t)$  is given by:

$$G(s) = \frac{bs + k}{s^2(mMs^2 + (M + m)bs + (M + m)k)} \quad (\text{a-1.1})$$

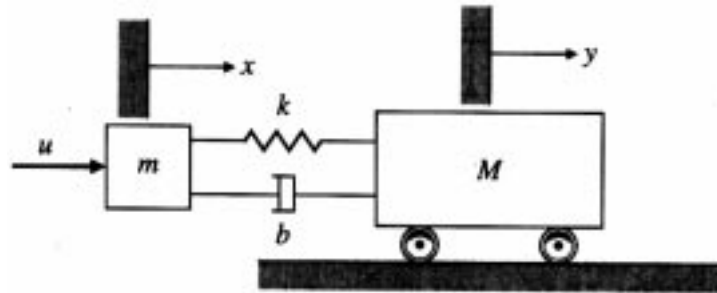


Fig. a-1.1. system configuration.

- a-1.2** Approximate  $G(s)$  assuming that  $m=0$ , and let  $M=1$ ,  $k=1$ ,  $b=0.1$ , and a compensator  $G_c(s) = K_p$  (i. e. proportional control action). Can  $K_p$  be chosen to satisfy the performance specifications? Why or why not?
- a-1.3** Suggest a compensator such that the performance specifications can be met.
- a-1.4** Plot the response of the closed loop system obtained in a-1.3 to step input.  
(Do not worry about the overshoot if it is bigger than 12% since the system is a HOS not a 2<sup>nd</sup> OS)
- a-1.5** *Uncertainty*: Now suppose that  $m$  is not negligible, and is given by  $m=M/10$ . Check to see if the controller you designed in 1-3 still meets the given specifications, if not, adjust the controller parameters so that the specifications are met. Plot the response of the obtained closed loop system to a step input.

#### Solution

- a-1.1** Derive the equation of motion of the system then apply Laplace transform (IC's=0) and simplify to get the transfer function  $G(s) = Y(s)/U(s)$ , as given in (a-1.1).
- a-1.2** Data:  $m=0$ , and let  $M=1$ ,  $k=1$ ,  $b=0.1$ , and  $G_c(s) = K_p$ .  
From the specifications we have:  $\omega_n = 1.8/t_r = 18 \text{ rad/s}$ ,  $z = 0.6$   
develop an m-file program to plot the RL of  $K_p G(s)$  as shown below.  
From the plot, the specification can not be met by just changing the gain  $k_p$   
The m-file used for this step and the output plot are given as follows (Fig. a-1.2):

m-file for question a-1.2

```
m=0;M=1;k=1;b=0.1;tr=0.1 % given data
ng=[b k];dg=[m*M (M+m)*b (m+M)*k 0 0]; % numerator and denominator of G
wn=1.8/tr % get wn from the given specification.
```

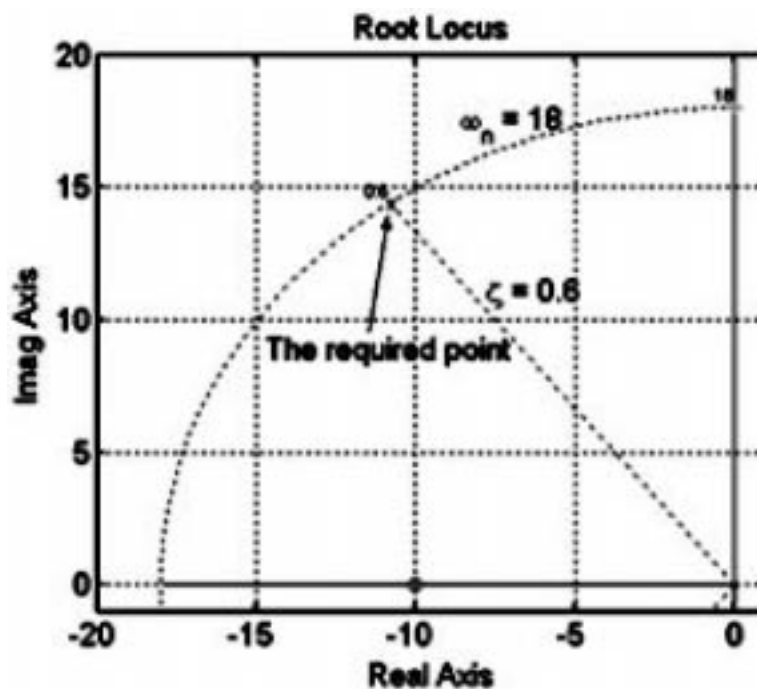


Fig. a-1.2. RL diagram for the P-control.

```

z=0.6 % the given damping ratio
rlocus(ng,dg);sgrid([z],[wn]); % Obtain the RL for KpG
axis([-20 1 -1 20]);
[k,poles]=rlocfind(ng,dg); % determine the poles and the corresponding gain

```

**a-1.3** It is required to propose a compensator such that the performance specifications can be met. Therefore, a lead compensator  $G_c(s) = \frac{s+z}{s+p}$  is used, where appropriate values of the zero  $z$  and pole  $p$  are ( $z=2.5$  and  $p=10z$ ), Fig. a-1.3 . The m-file used for this part is given as follows:

*m-file for question a-1.3*

```

clear all;figure;
m=0;M=1;k=1;b=0.1;tr=0.1 % given data
ng=[b k];dg=[m*M (M+m)*b (m+M)*k 0 0]; % numerator and denominator of G
z=2.5;p=10*z;ngc=[1 z];dgc=[1 p]; % numerator and denominator of Gc
G=tf(ng,dg);Gc=tf(ngc,dgc); % Obtain the transfer functions of G and Gc
nggc=conv(ng,ngc);dggc=conv(dg,dgc); % numerator and denominator of the CLS
wn=1.8/tr % get wn from the given specification.
zeta=0.6 % the given damping ratio
rlocus(nggc,dggc);sgrid([zeta],[wn]); % Obtain the RL for GGc
axis([-20 1 -1 20]);

```

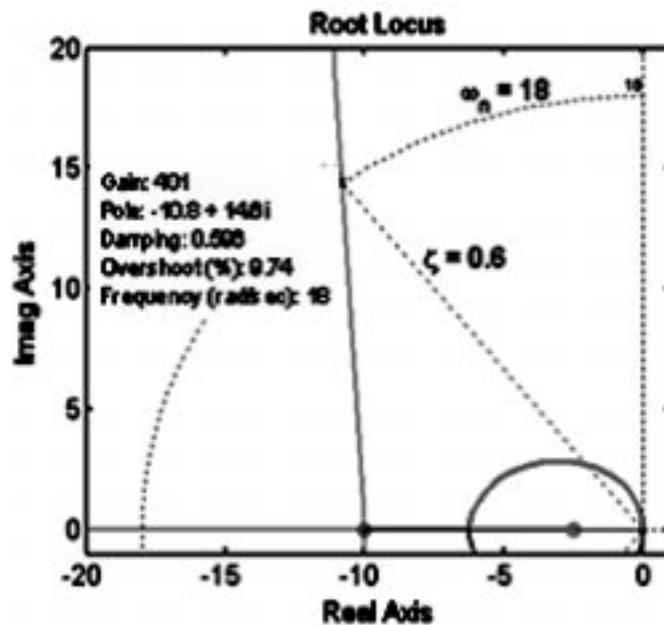


Fig. a-1.3. RL diagram for the Lead/Lag compensator control.

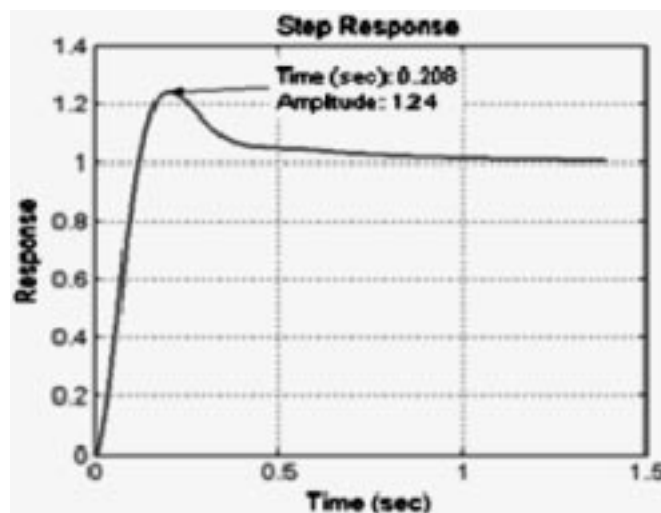


Fig. a-1.4. Step input response for the system with Lead/Lag compensator control.

**a-1.4** Plot the response of the closed loop system obtained in part a-1.3 to step input. Fig. a-1.4 shows the plot together with the answer. (Do not worry about the overshoot if it's bigger than 12% since the system is a HOS)

*m-file for question a-1.4*

```
clear all;figure;
m=0;M=1;k=1;b=0.1;tr=0.1 % given data
ng=[b k];dg=[m*M (M+m)*b (m+M)*k 0 0]; % numerator and denominator of G
z=2.5;p=10*z;ngc=[1 z];dgc=[1 p]; % numerator and denominator of Gc
G=tf(ng,dg);Gc=tf(ngc,dgc); % Obtain the transfer functions of G and Gc
nggc=conv(ng,ngc);dggc=conv(dg,dgc); % numerator and denominator of the CLS
tf1=tf(401*nggc,dggc);tf2=tf([1],[1]); % Obtain the transfer functions of the CLS at
the % selected CLPoles
step(feedback(tf1,tf2));grid % Get the response to a unit step command input
```

**a-1.5** *Uncertainty*: Now let us investigate the uncertainty, the value of  $m = 0.1M$  (see (a-1.1)) is assigned as shown in the m-file. Running the program, the specifications are not met because of the two complex poles initiated when the uncertainty is considered. Therefore, a notch filter having complex zeros (very close to the unwanted complex poles) and two real poles far away from the j-axis is serially added to the lead compensator (Fig. a-1.5).

*m-file for question a-1.5*

```
clear all;figure;
m=0.1;M=1;k=1;b=0.1;tr=0.1; % the given data with uncertainty
ng=[b k];dg=[m*M (M+m)*b (m+M)*k 0 0]; % numerator and denominator of G
z=2.5;p=5*z;ngc=[1 z];dgc=[1 p]; % numerator and denominator of Gc
notch=zpk([-0.5+3i,-0.5-3i],[-40,-40],1) % Notch filter
G=tf(ng,dg);Gc=tf(ngc,dgc)*notch; % Obtain the feedforward transfer functions.
wn=1.8/tr % get wn from the given specification.
zeta=0.6 % the given damping ratio
rlocus(G*Gc);sgrid([zeta],[wn]); % Obtain the RL for GGc
axis([-20 4 -1 20]);
figure;
tf2=tf([1],[1]);
step(feedback(1.61*10^4*G*Gc,tf2)) % Get the response to a unit step command input
```

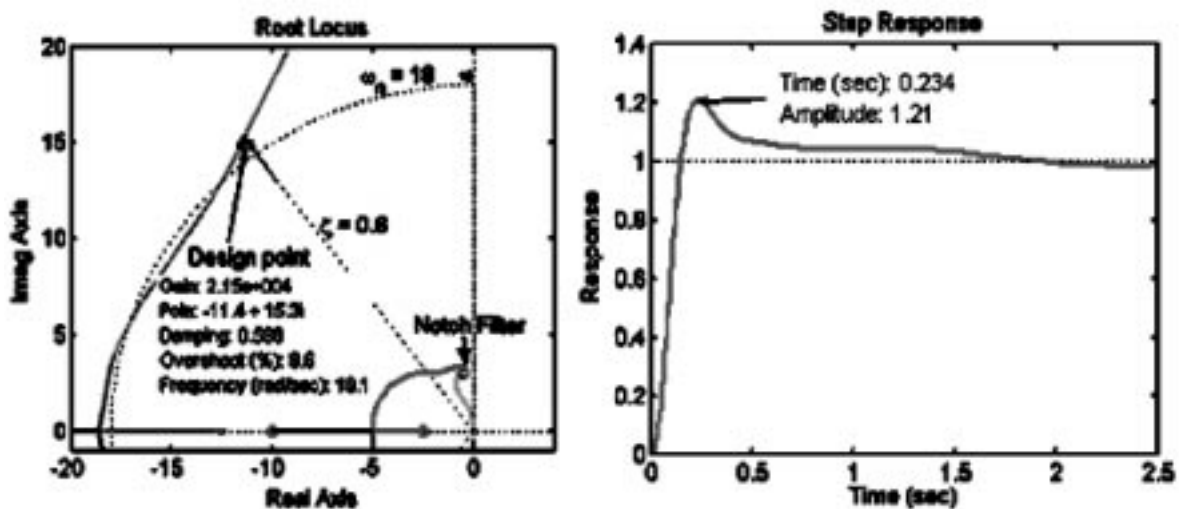


Fig. a-1.5. RL and step input response for the system with Lead/Lag and notch compensator.

APPENDIX 2

Example a-2

A second-order system is represented by the differential equation:

$$m\ddot{y} + c\dot{y} + ky = F, \tag{a2.1}$$

For  $m = 1$  kg,  $c = 2$  Ns/m,  $k = 4$  N/m; and  $F = \sin 1.4t$  N, and with zero initial conditions.

Suppose that the force  $F$  is created by applying a sinusoidal input voltage to hydraulic piston that has dead-zone nonlinearity of  $\pm 0.5$  s. This means that the piston does not generate a force until the input voltage exceeds a certain magnitude [9].

**a-2.1** Model and simulate the system using Simulink.

**a-2.2** Plot the time- displacement response of the system and the time- applied force on the system.

**a-2.3** Plot the phase portrait of the System and comment on its significant.

Solution

**a-2.1** To model the system let us rewrite (a2.1) as:

$$\ddot{y} = -\frac{c}{m}\dot{y} - \frac{k}{m}y + \frac{1}{m}F \tag{a2.2}$$

Equation (a2.2) can be represented as a block diagram in Simulink as shown in Fig. (a-2.1).

**a-2.2 and a-2.3** the Simulink program was run and the output yy is exported to the workspace and an m-file was developed to plot the graphs as shown in Fig. a-2.2.

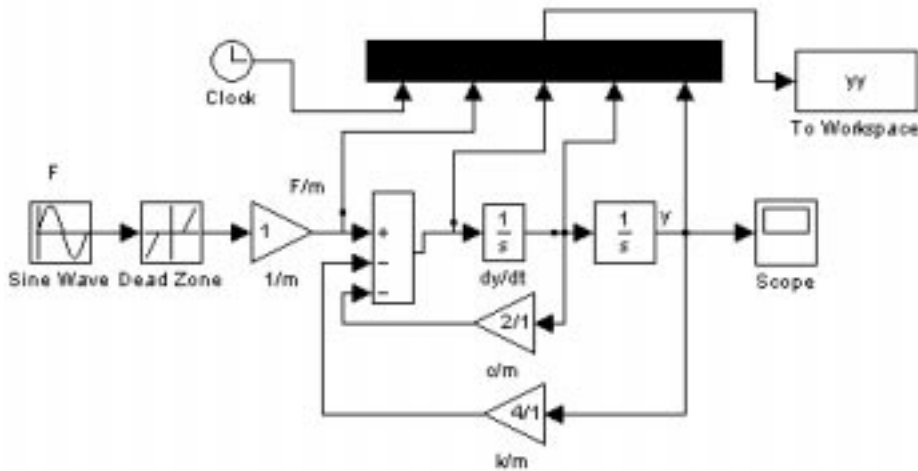


Fig. a-2.1. Modeling of system (a-2.1) using Simulink.

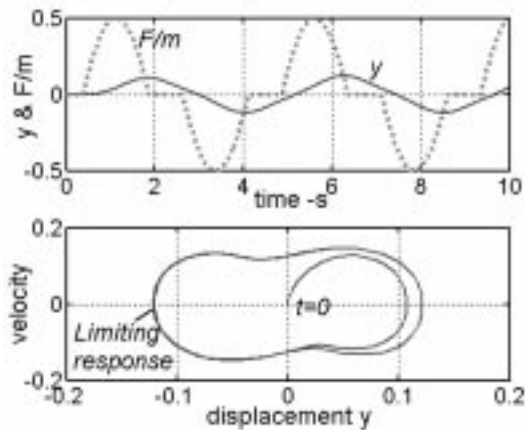


Fig. a-2.2. The time versus displacement response and input force (top), and phase portrait (bottom).



As shown in Fig. a-2.2 (lower), since there is no initial conditions, the displacement-velocity response is equal to zero at  $t=0$ . The response reaches a limiting trajectory at steady state where the energy added to the system by the forcing function is equal to the energy dissipated by the damping.

**Mostafa S. Habib** is from Egypt and is currently an associate professor with the department of mechanical engineering, university of Bahrain, Kingdom of Bahrain. He obtained the B.Sc. and M.Sc. degrees in mechanical engineering both from Ain Shams University, Cairo, Egypt in 1973 and 1978 respectively. He obtained M.Sc. in mathematics and Ph.D. in vibration/control both from Michigan State University, East Lansing, Michigan, USA in 1986 and 1988 respectively. His current research interests include the use of software facilities in teaching mechanical engineering courses and also the control applications in the field of vehicle-human protection.