5-2015

GUI Matlab code to display damped, undamped, forced and unforced mass spring systems

Melanie Garcia
University of Arkansas, Fayetteville

GUI MATLAB CODE TO DISPLAY DAMPED, UNDAMPED, FORCED AND UNFORCED MASS SPRING SYSTEMS

GUI MATLAB CODE TO DISPLAY DAMPED, UNDAMPED, FORCED AND UNFORCED MASS SPRING SYSTEMS

An Undergraduate Honors College Thesis

In the

Department of Mechanical Engineering

College of Engineering

University of Arkansas

Fayetteville AR

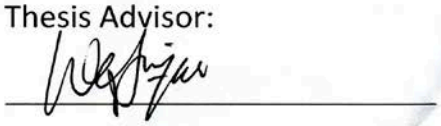
By

Melanie Garcia Camberos

April 20 2015

This thesis is approved.

Thesis Advisor:



Thesis Committee:

ACKNOWLEDGEMENTS

I would like to thank Dr. Wejinya for all his help and patience during the composition of this project. Thank you Dr. Wejinya for coming up with a project for me in such short notice and for the guidance provided. I would also like to thank Dr. Roe for taking time to listen to my presentation.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Machine Dynamics and Control is the course offered at the University of Arkansas that deals with damped, undamped, forced and unforced systems. It's important for students to understand each of these systems for future implementation in the industry and daily real life situations. The three types of damped systems, underdamped, overdamped and critically damped are analyzed in this thesis. As well as only two of the most common forced functions are analyzed, the cosine and sine functions. The objective of this thesis was to develop a GUI code in Matlab that would help students visualize the differences between undamped, damped, forced and unforced mass spring systems. It would also create an easy way for students to solve the problems assigned from the textbook so that a different Matlab code would not have to be created every time. At the end a display window was created were the input can be used to create the graph of the total response vs time which displays the oscillation of the system with the values of the integration constants beside it.

I. INTRODUCTION

A. General Introduction

The course Machine Dynamics and Control at the University of Arkansas deals with undamped, damped, forced and unforced mass spring systems. The energy equation is the basis from where all the total response equations and integrated constants are derived from. The undamped and damped systems have a strong differentiation in their oscillation that can be better understood by looking at their graphs side by side. There are three different types of damped systems: underdamped, overdamped and critically damped. Each of these three types has their own properties that makes them applicable in different situations. The forced and unforced systems also differ from each other in that a forced system is excited by a force applied to it and in this case only the sine and cosine forcing functions are going to be explained. The total response for each system is graphed versus time to view the oscillation of the base. All of this information can be applied to real world situations like the damping of a car, which is why its important for students to really understand it.

B. Research Problem

Students have a hard time understanding the differences between the systems named above. The biggest problem they face is depicting the difference between an undamped and damped system. The GUI code in this project will enable students to input values from different textbook problems and have a visual of what the graph for the oscillation looks like. This can serve as a visualization tool for the problem and also as a way to check if the problem they are solving is correct. Another problem faced when solving the mass spring system is that every time a different type of problem wants to be solved (forced, unforced, damped or undamped) a new set of code needs to be created because each system has its own total response equation. With the code developed below none of this has to be done; only the right input values have to be inserted and the program will do the rest.

II. NOMENCLATURE

m	Mass of object
c	Damping coefficient
k	Spring constant

t	Time
f(t)	Forcing function
ζ	Damping Ratio
fo	Initial Forced Function/Mass
Fo	Initial Forced Function
w	Frequency
wn	Natural frequency
wd	Damping frequency
xo	Initial displacement
vo	Initial velocity
ϕ	Constants of integration
θ	Phase Constants of integration
A	Amplitude Constants of integration
X	Magnitude

III. LITERATURE REVIEW

A. Undamped vs. Damped Systems

The equation of motion is going to be used throughout this research so it's important to understand how to derive it from Fig. 1.

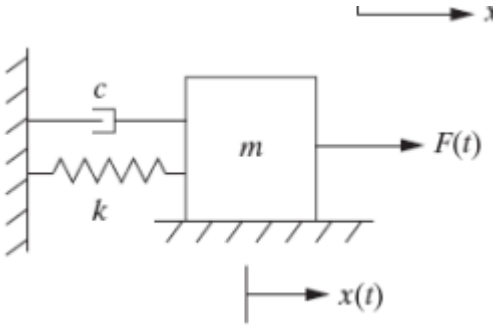


Fig.1: Spring System [2]

The free body diagram for the spring system (see Fig. 2) can then be used to derive the equation.

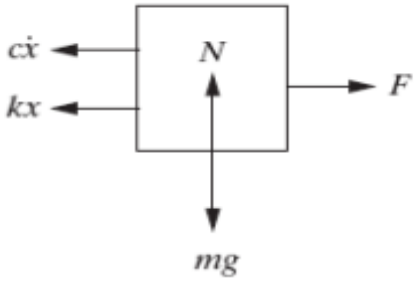


Fig. 2: Free Body Diagram of Spring System [2]

Adding the horizontal forces we get Eq. 1. The vertical forces are also added up but they are negligible because the mass is only moving horizontally.

$$-kx - c\dot{x} + f(t) = m\ddot{x} \qquad \textit{Eq. 1}$$

Rearranging the variables in Eq.1 we get to the equation of motion.

$$m\ddot{x} + c\dot{x} + kx = f(t) \qquad \textit{Eq. 2}$$

For an undamped system the damping coefficient (c=0) is equal to zero. The graph caused by this characteristic can be seen in Fig. 3.

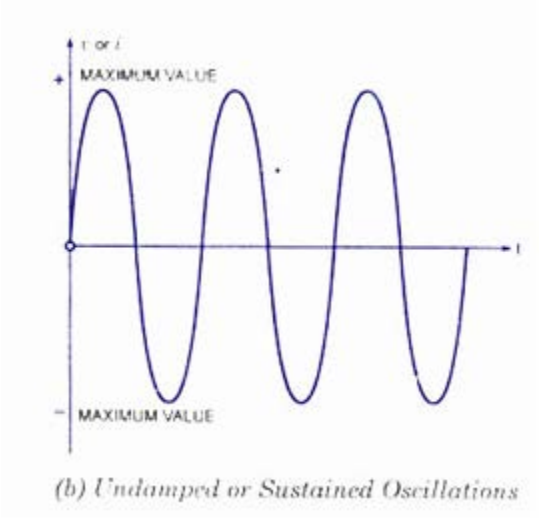


Fig. 3: Undamped Oscillations Graph [4]

The general response for the free response undamped case has the form of Eq. 3. The equations for the forced response of the undamped system will be explained in a later section.

$$x(t) = A * \sin(wn * t + \phi) \qquad Eq.\ 3$$

The natural frequency is represented by wn and can be calculated with Eq.4.

$$wn = \sqrt{\frac{k}{m}} \qquad Eq.\ 4$$

We can then calculate A and ϕ based on the initial conditions using Eq. 5 and Eq. 6.

$$A = \frac{\sqrt{wn^2 * xo^2 + vo^2}}{wn} \qquad Eq.\ 5$$

$$\phi = \arctan(\frac{wn * xo}{vo}) \qquad Eq.\ 6$$

To generate Fig. 3 the general response with its units in meters is graphed against time.

On the other hand, the damped system has a value assigned for the damping coefficient that depends on the value of the mass, spring constant and damping ratio. The damping ratio is a number bigger than 0 that depends on if the system is critically damped, overdamped or underdamped. The damping cooefficient can be calculated using Eq. 7.

$$c = 2 * m * wn * \zeta \qquad Eq.\ 7$$

The graph for a damped system depends on the value of the damping ratio which in turn affects the damping coefficient. The graph in Fig. 4 shows a standard damping system. The general response for the underdamped, critically damped and overdamped will be analyzed in the next section.

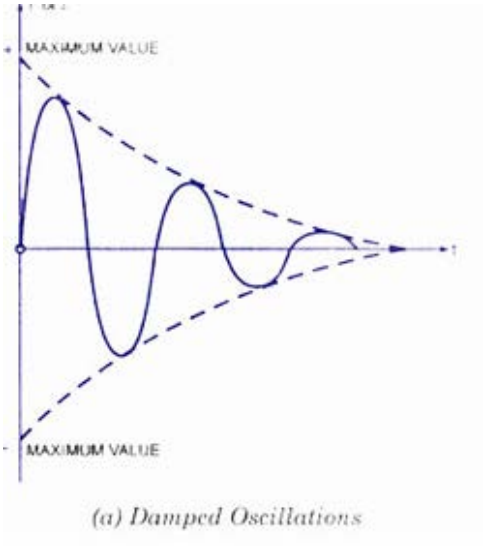


Fig. 4: Damped Oscillations Graph [4]

As seen from Fig. 3 and Fig. 4 the undamped system oscillates into infinity while the damping system has a time constant where the oscillation ends. The time constant is unique for each type of damping system and situation.

B. Types of Damped Systems

a) Underdamped

For an underdamped system the damping ratio is between zero and one ($0<\zeta<1$). This is the most common case and the only one that yields oscillation as seen in Fig. 5. The underdamped system gives an oscillation response with an exponential decay. Most of the natural systems vibrate in this fashion.

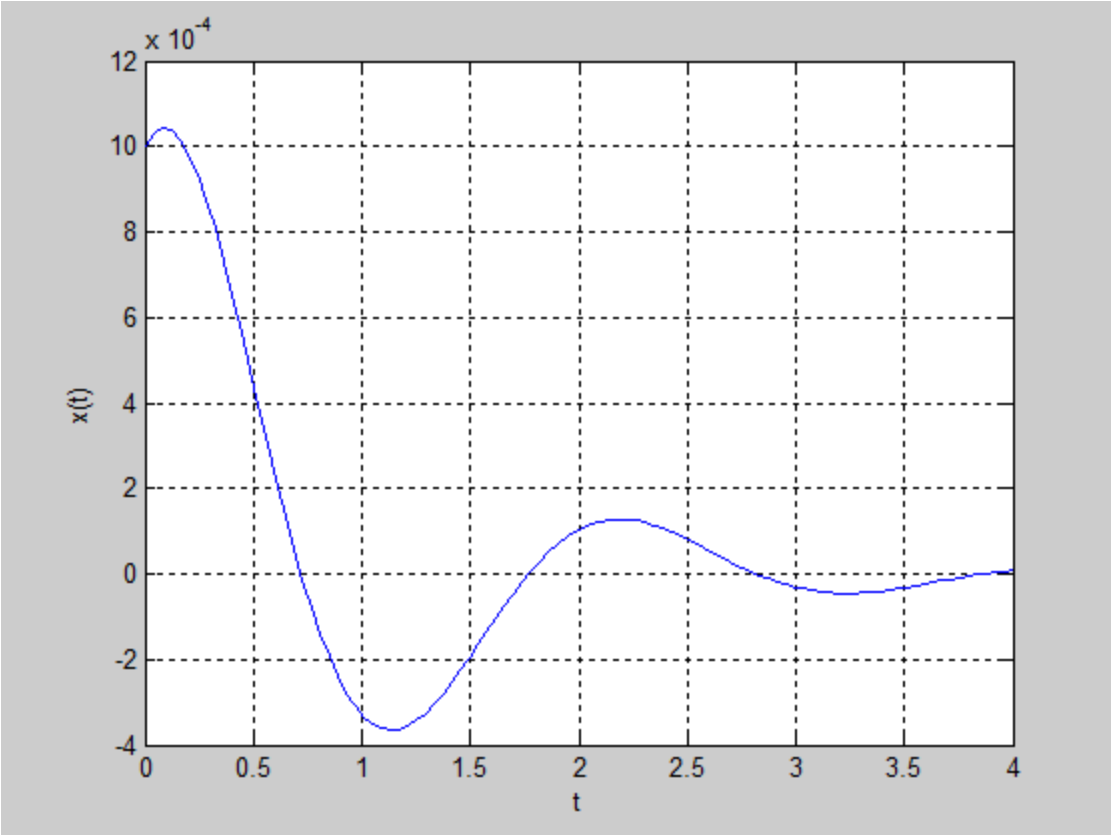


Fig. 5: Underdamped System Question 1.41 [1]

Underdamped oscillation has its own frequency of oscillation called the damping frequency which can be calculated by Eq. 8.

$$wd = wn\sqrt{1-\zeta\text{^}2} \qquad Eq.\ 8$$

The graph above was created by plotting the general response in Eq. 9 versus time.

$$x(t) = A * e^{-\zeta * wn * t} * \sin(wd * t + \phi) \qquad Eq.\ 9$$

The natural frequency and the damping ratio can be calculated using Eq. 4 and Eq. 7 respectively. We can then calculate A and ϕ based on the initial conditions using Eq. 10 and Eq. 11.

$$\phi = \arctan\left(\frac{xo * wd}{vo + \zeta * wn * xo}\right) \qquad Eq.\ 10$$

$$A = \sqrt{\frac{(vo + \zeta * wn * xo)^2 + (xo * wd)^2}{wd^2}} \qquad Eq.\ 11$$

Fig. 5 was taken from question number 1.41 in the book Engineering Vibration and will be compared in a later section to the graph obtained from the GUI. For the calculations and values of mass, spring constant and damping coefficient refer to Appendix A.

b) Overdamped

In an overdamped system the damping ratio is greater than 1 (δ>1). The general response to this system is shown in Eq. 12 and this is graphed versus time in Fig. 6.

$$x(t) = e^{-\zeta * wn * t} * (a1 * e^{-wn * t * \sqrt{\zeta^2 - 1}} + a2 * e^{wn * t * \sqrt{\zeta^2 - 1}}) \qquad Eq.\ 12$$

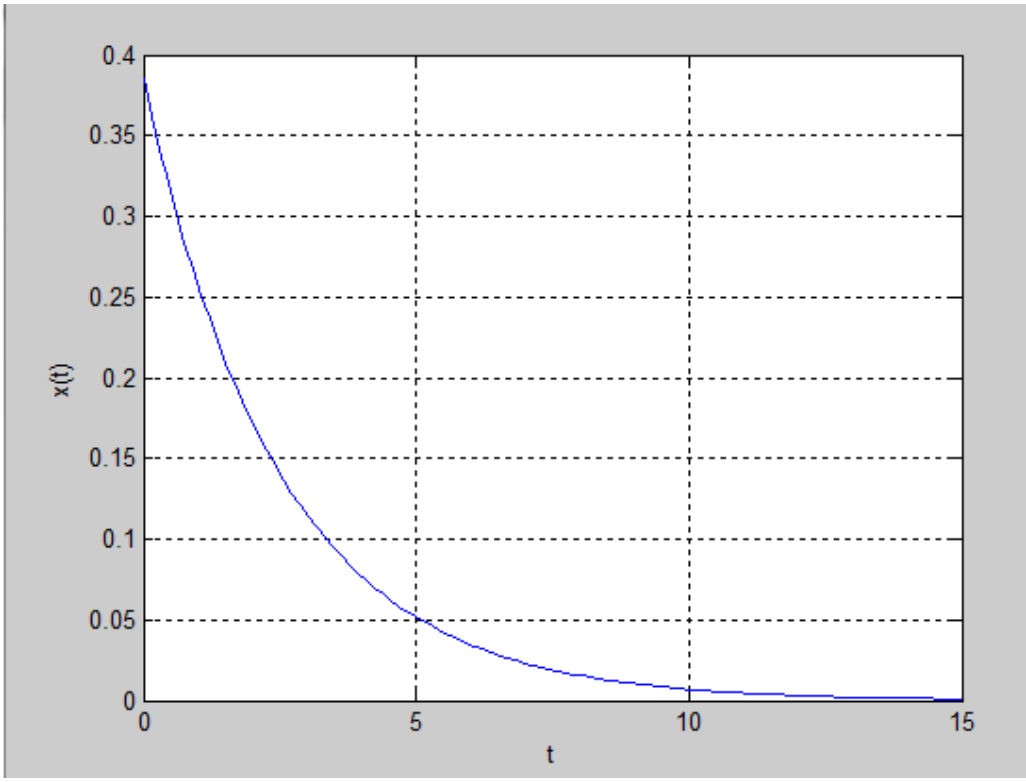


Fig. 6: Overdamped System 1.43 [1]

From Fig. 6 it is clear that an overdamped system doesn't oscillate and it returns to its rest position exponentially. [1] It is also slower to respond than a critically damped system in Fig. 7. The natural frequency and the damping ratio can be calculated using Eq. 4 and Eq. 7 respectively. From the initial conditions, a1 and a2 can be calculated with Eq. 13 and Eq.14.

$$a1 = \frac{-vo+\left(-\zeta+\sqrt{\zeta^2-1}\right)*wn*xo}{2*wn*\sqrt{\zeta^2-1}} \qquad \textit{Eq. 13}$$

$$a2 = \frac{vo+\left(\zeta+\sqrt{\zeta^2-1}\right)*wn*xo}{2*wn*\sqrt{\zeta^2-1}} \qquad \textit{Eq. 14}$$

Fig. 6 was taken from question 1.43 in the book Engineering Vibration and will be compared in a later section to the graph obtained from the GUI. For the calculations and values of mass, spring constant and damping coefficient refer to Appendix B.

c) Critically Damped

For a critically damped system the value of the damping ratio is equal to 1 (ζ=1). The general response for this system is shown in Eq. 15 and it is graphed versus time in Fig. 7

$$x(t) = (a1 + a2 * t) * e^{-wn*t} \qquad Eq.\ 15$$

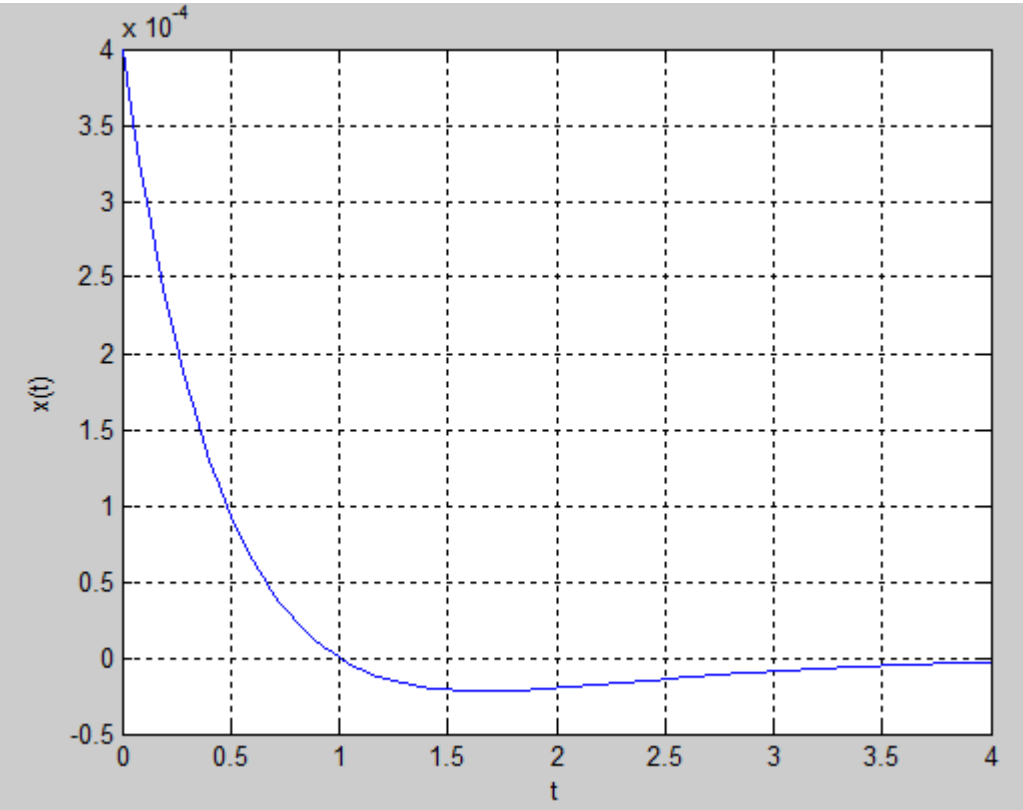


Fig. 7: Critically Damped System Example 1 [1]

In a critically damped system no oscillation occurs. It has the value of damping that provides the fastest return to time zero without oscillation. [1].

The natural frequency can be calculated using Eq. 4. From the initial conditions a1 and a2 can be calculated using Eq. 16 and Eq. 17.

$$a1 = xo \qquad Eq.\ 16$$

$$a2 = vo + wn * xo \qquad Eq.\ 17$$

Fig. 7 was taken from an example in the class notes from Dr. Wejinya, it will be compared in a later section to the graph obtained from the GUI. For the calculations and values of mass, spring constant and damping coefficient refer to Appendix C.

C. Unforced Damped Systems

An unforced damped system consists of the equation of motion (Eq. 2) equal to zero. Therefore, Eq. 2 becomes the following

$$m\ddot{x} + c\dot{x} + kx = 0 \qquad Eq.\ 18$$

For all the cases described above (underdamped, overdamped and critically damped) it was assumed that they were all unforced damping systems.

D. Forced Damped Systems

For a forced damped system the equation of motion looks like Eq. 2 and for f(t) a variety of different forcing functions can be applied. For the purpose of this research only two forcing functions where considered, cosine and sine.

a) Cosine

Fig. 8 below demonstrates how the cosine forced function is applied to a mass spring system, which is shown by Eq. 19.

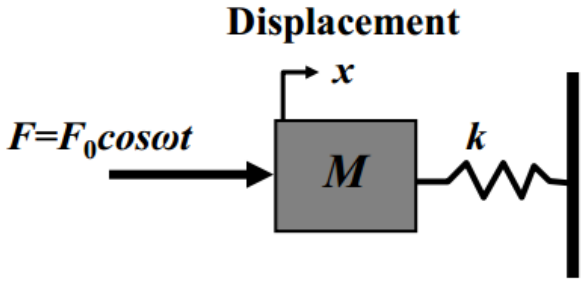


Fig. 8: Cosine Force Applied to Mass Spring System [2]

$$F(t) = Fo * cos(wt) \qquad Eq.\ 19$$

We can divide Eq. 19 by the mass to get fo in Eq. 20.

$$fo = \frac{Fo}{m} \qquad Eq.\ 20$$

First we start with the undamped situation, which means the damping coefficient is equal to zero (c=0). The solution to this force is the sum of the particular and homogenous solutions. The particular solution for the cosine force can be seen in Eq. 21.

$$x_p(t) = X * \cos(wt) \qquad Eq.\ 21$$

Adding the homogeneous solution from Eq.3 to Eq. 21 gives the total solution in Eq. 22 for an undamped system.

$$x(t) = A * \sin(wn * t + \phi) + X * \cos(w * t) \qquad Eq.\ 22$$

The constants of integration X, ϕ and A can be calculated using Eq. 23, 24 and 25 that depend on the initial conditions.

$$X = \frac{fo}{wn^2 - w^2} \qquad \textit{Eq. 23}$$

$$\phi = \arctan\left(\frac{wn*(xo-X)}{vo}\right) \qquad \textit{Eq.24}$$

$$A = \sqrt{\left(\frac{vo}{wn}\right)^2 + (xo - X)^2} \qquad \textit{Eq. 25}$$

The same logic applies to the forced underdamped system. The homogenous solution in Eq. 9 is added with a different particular solution in Eq. 26 to produce the total solution in Eq. 27.

$$x_p(t) = X * \cos(wt - \theta) \qquad \textit{Eq. 26}$$

$$x(t) = A * e^{-\delta*wn*t} * \sin(wd * t + \phi) + X * \cos(w * t - \theta) \qquad \textit{Eq. 27}$$

For this case the integration constants still depend on the initial conditions, but are calculated with different equations (Eq. 28,29, 30 and 31).

$$X = \frac{fo}{\sqrt{(wn^2-w^2)^2+(2*\zeta*wn*w)\text{^}2}} \qquad \textit{Eq. 28}$$

$$\theta = \arctan\left(\frac{2*wn*w*\zeta}{wn^2-w^2}\right) \qquad \textit{Eq. 29}$$

$$\phi = \arctan\left(\frac{wd*(xo-Xcos\theta)}{vo+(xo-X*cos\theta)*\delta*wn-w*X*sin\theta}\right) \qquad \textit{Eq. 30}$$

$$A = \frac{xo-X*cos\theta}{sin\phi} \qquad \textit{Eq. 31}$$

As seen from the equations above the forcing function in the particular solution is affected by the amplitude and phase. If the natural frequency and the forcing frequency are close together then the amplitude of the particular solution will be very large [2].

b) Sine

The sine forcing function would be exactly the same as Fig. 8, but instead of cosine the force would have a sine. The solution for this system can be calculated in the same way as the cosine total solution was calculated, starting with the particular solution in Eq. 32.

$$x_p(t) = X * \sin(w * t) \qquad Eq.\ 32$$

For the undamped situations Eq. 33 shows the total solution.

$$x(t) = A1 * \sin(wn * t) + A2 * \cos(wn * t) + X * \sin(w * t) \qquad Eq.33$$

The value of X is calculated with Eq. 23, the same equation used for the cosine forcing function. The other integration constants are calculated with Eq. 34 and Eq. 35.

$$A1 = \frac{vo}{wn} - \frac{w}{wn} * \frac{fo}{wn^2 - w^2} \qquad Eq.\ 34$$

$$A2 = xo \qquad Eq.\ 35$$

When the sine forced function is underdamped the equation is similar as Eq. 27 for an underdamped cosine forced function. The sine function lags the cosine function and this can be applied to Eq. 27 to obtain the total response for the sine function. By subtracting π/2 from every angle in Eq. 27, we get the sine total response in Eq. 36.

$$x(t) = A * e^{-\zeta * wn * t} * \sin\left(wd * t + (\phi - \frac{\pi}{2})\right) + X * \cos(w * t - (\theta - \frac{\pi}{2})) \qquad Eq.\ 36$$

The integration constants are still calculated by Eq. 28, 29, 30 and 31.

IV. SYSTEM DEVELOPMENT

A GUI was created in Matlab that will help students interact with the program by inputting different values. The values that the student can input are seen in Fig. 9.

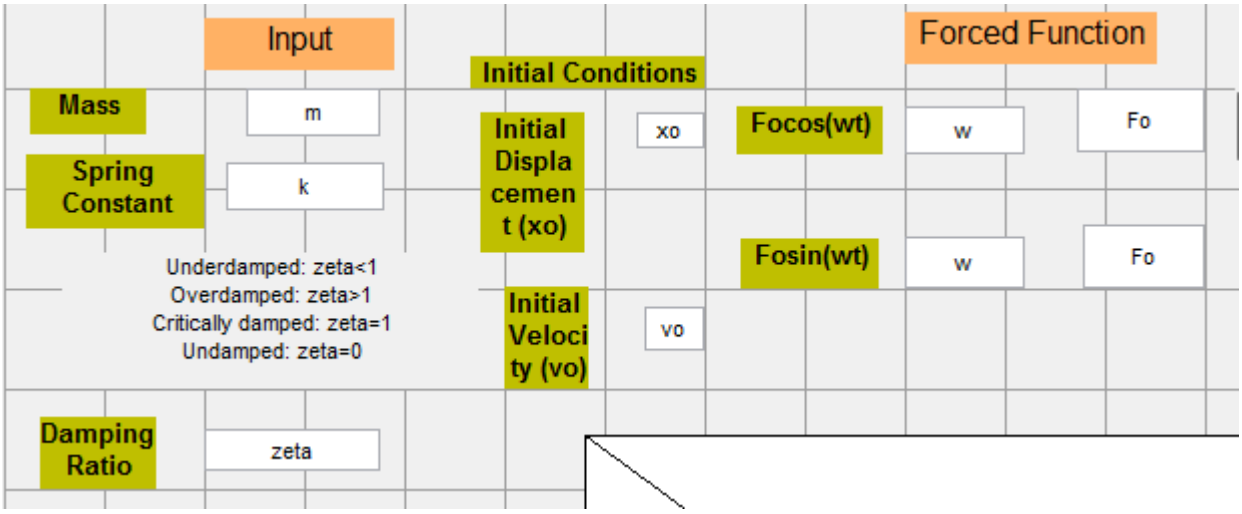


Fig. 9: Input Values

Then the input values are extracted and used in the following calculations (See Fig. 10).

```
%gathering input
m=str2num(get(handles.mass,'String'));
k=str2num(get(handles.spring_constant,'String'));
zeta=str2num(get(handles.damping_ratio,'String'));
xo=str2num(get(handles.xo,'String'));
vo=str2num(get(handles.vo,'String'));

wn=sqrt(k/m);
set(handles.natural_frequency,'String',wn)
c=zeta*(2*m*wn);
set(handles.damping_coefficient,'String',c)
```

Fig. 10: Input code

The input value of the damping ratio is evaluated with an if loop to check if it's underdamped ($0<\zeta<1$), overdamped ($\zeta>1$) or critically damped ($\zeta=1$).

The output also shows the value for the integration constants as seen in Fig. 11.

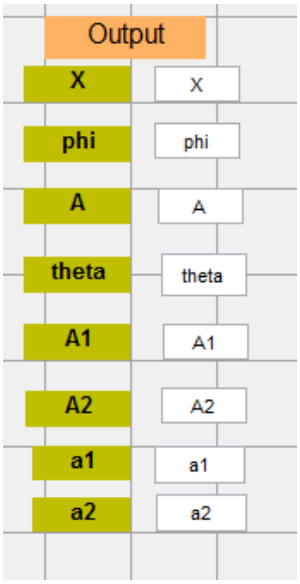


Fig. 11: Output of integration constants

The value of A is the amplitude of the graph and by outputting the value of it we can depict how big the amplitude will be and associate it with the graph. The variable X depicts the magnitude of the forced function acting on the system. Phi is the phase change the graph undergoes between the undamped and the damped system.

A. Undamped System

When the for loop detects that the damping ratio is 0 it goes directly to Eq. 3, 5 and 6 to calculate the response of the system. Pressing the graph unforced bottom Eq. 3 is plotted versus time. As an example of how the program works, question 1. 19 from the text book Engineering

Vibrations will be plotted. The input values, output values and the graph for the undamped example can be seen in Fig. 12.

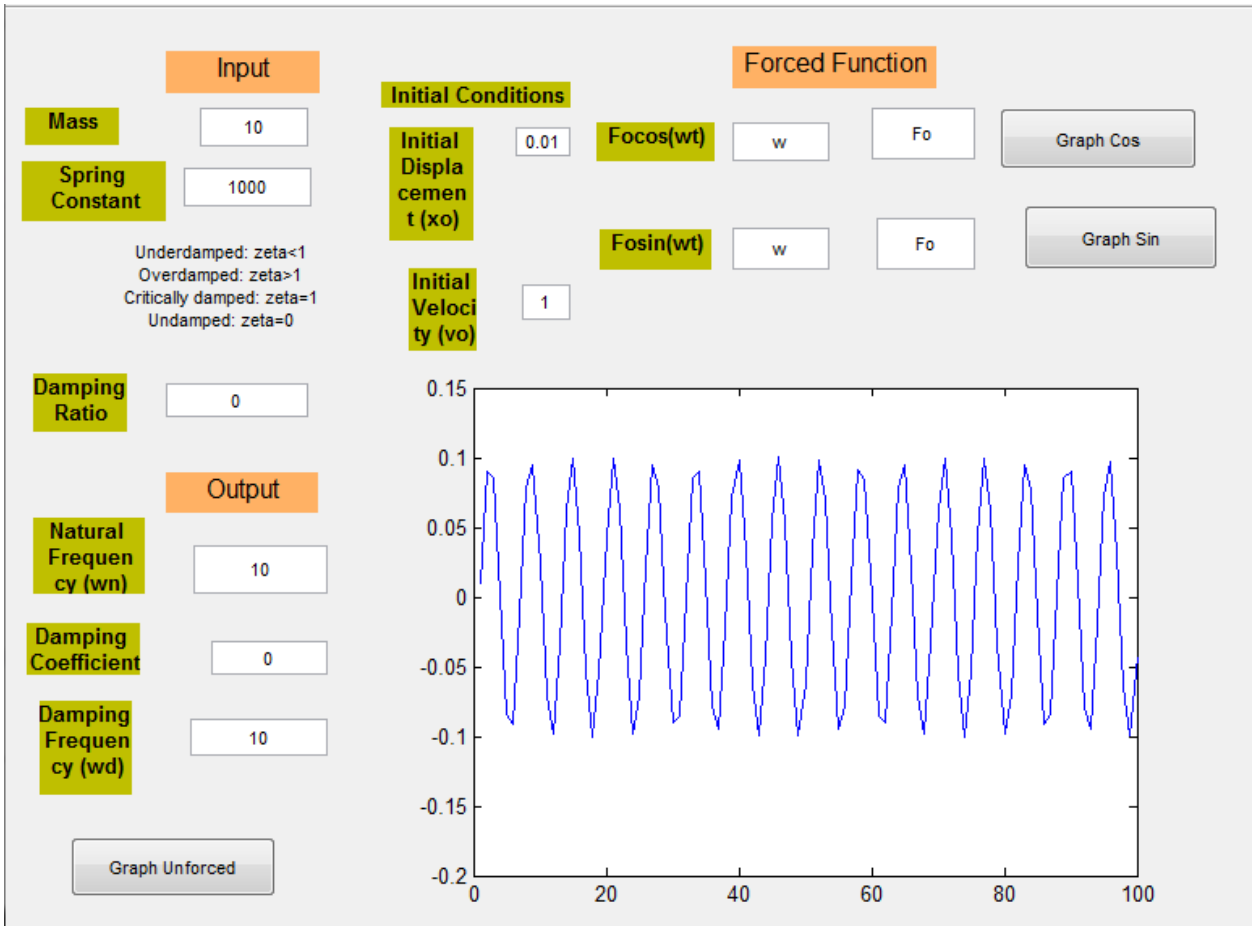


Fig.12: Undamped System Question 1.19 [1]

The detailed code and the question can be found in Appendix D. Comparing the experimental graph in Fig. 12 to the theoretical graph in Fig. 3 we can see that both graphs are similar and oscillate to infinity. Thus we can conclude that the code is working right because of this similarity.

B. Damped Systems

a) Underdamped

For the underdamped system as stated before the damping ratio needs to be between zero and one. When the for loop detects that this is the case, the values for Eq. 8, 9 ,10 and 11 are calculated with the code in Appendix E. When the graph unforced button is pushed, a graph of the total response in Eq. 9 versus time is produced as seen in Fig. 13.

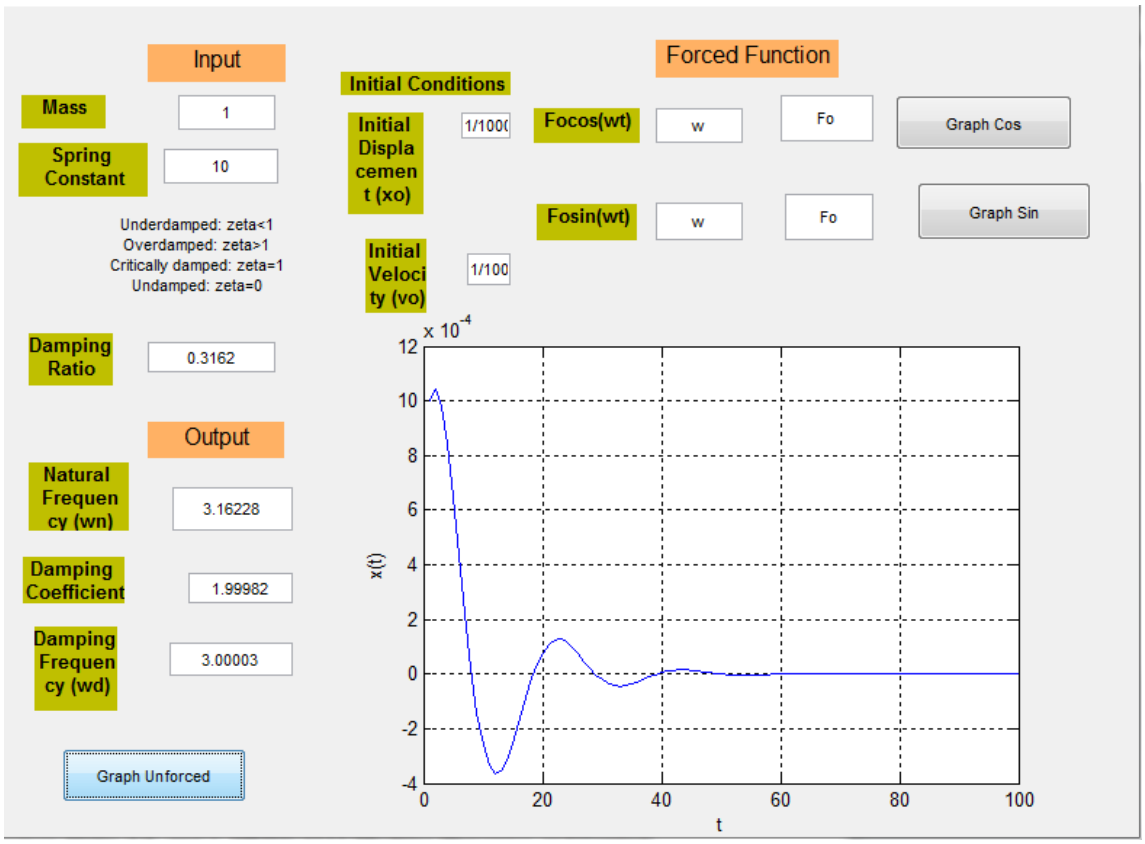


Fig.13:Underdamped Display for Question 1.41 [1]

Question 1.41 from the textbook Engineering Vibrations was used to produce the graph in Fig. 13. The detailed question and code can be found in Appendix E. Comparing the experimental graph in Fig. 13 to the theoretical graph in Fig. 5 we can see that they are exactly the same and that around time 60 seconds the system stops to oscillate.

Another feature of the code is that the underdamped and undamped graph can be seen at the same time and side by side. This helps the student visualize the difference between the two graphs which is the major problem the students have. Fig. 14 shows the underdamped and undamped graphs for question 1.41. The way the code works is first the underdamped data has to be entered and then the damping ratio value is converted to zero and that's how the undamped graph is created.

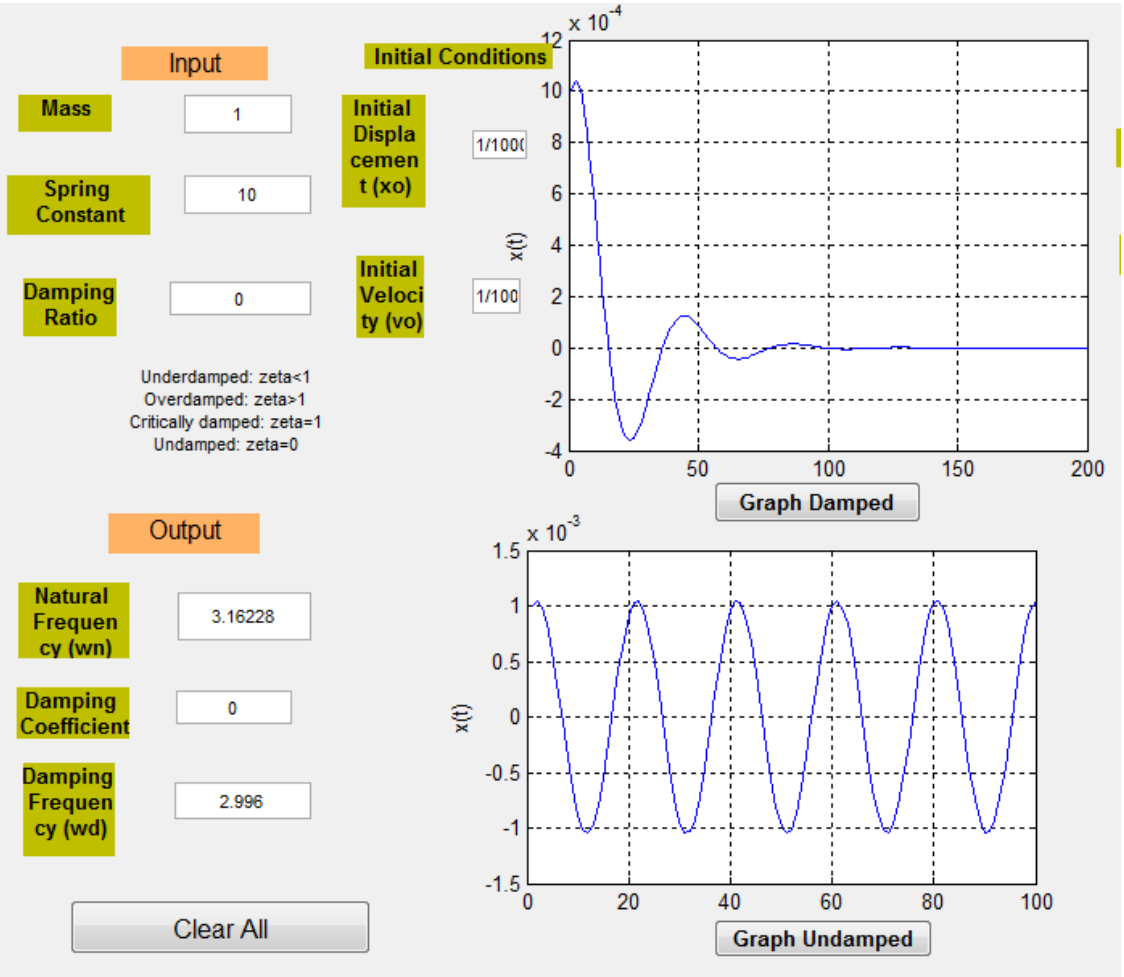


Fig. 14: Undamped and Underdamped Question 1.41

b) Overdamped

For an overdamped system the damping ratio is bigger than 1 ($\zeta>1$) as stated before. The total response is obtained from Eq. 12 with the integration constants calculated by Eq. 13 and 14. After entering the input values one presses the graph unforced bottom to produce the graph in Fig. 14.

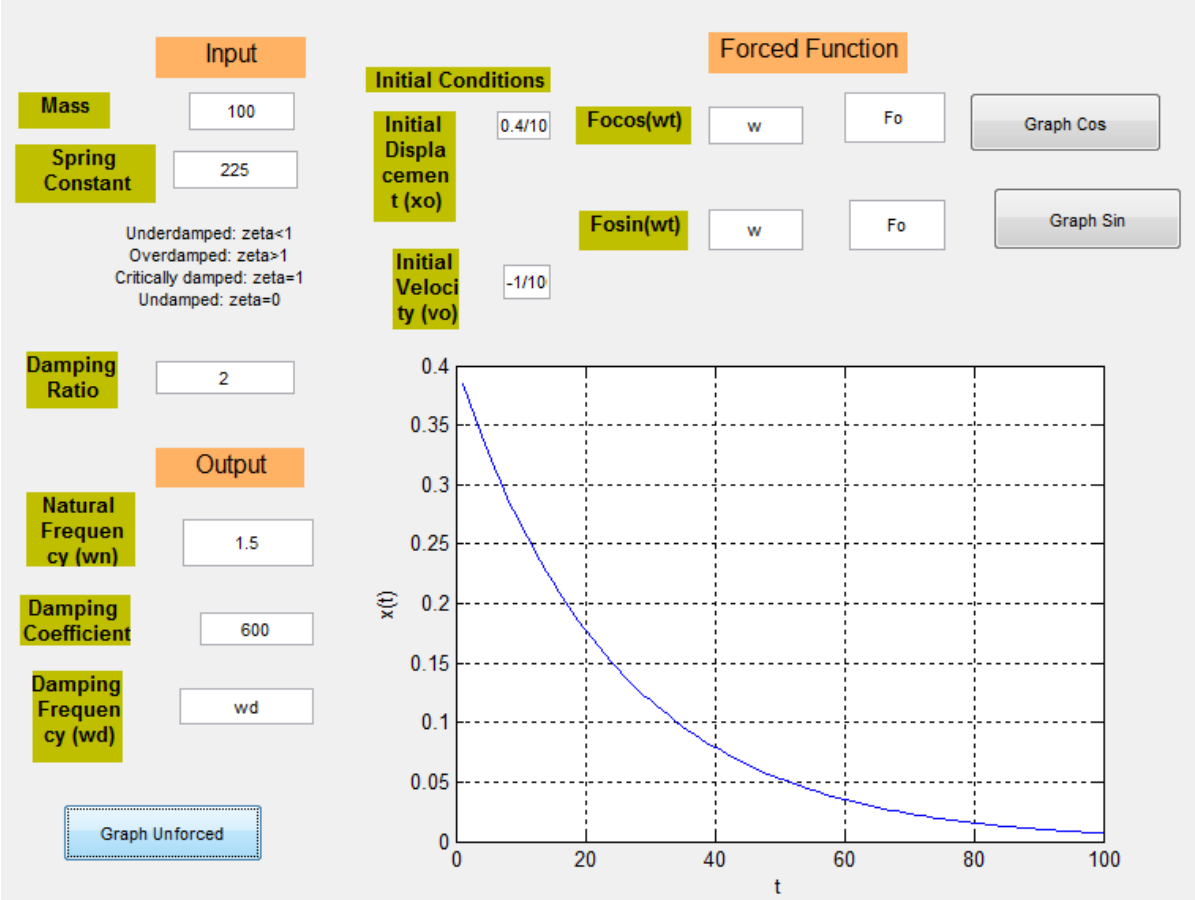


Fig.14: Overdamped System Display for Question 1.43

The display screen above was generated from question 1.43 in the book Engineering Vibrations. The detailed input and code can be found in Appendix E. Comparing the graph above with Fig.6 we see that they are identical further proving that the GUI is accurate. From Fig. 14 we can also see that an overdamped system doesn't oscillate at all, it drops directly to zero.

c) Critically Damped

For a critically damped system the damping ratio is equal to 1 (ζ=1) as stated before. The same inputs as the previous two examples are needed to be able to calculate the overall system response. These inputs are then plugged into Eq. 15, 16 and 17 to calculate the overall system response. When the graph unforced button is pushed the graph of Eq. 12 versus time is created as seen in the figure below.

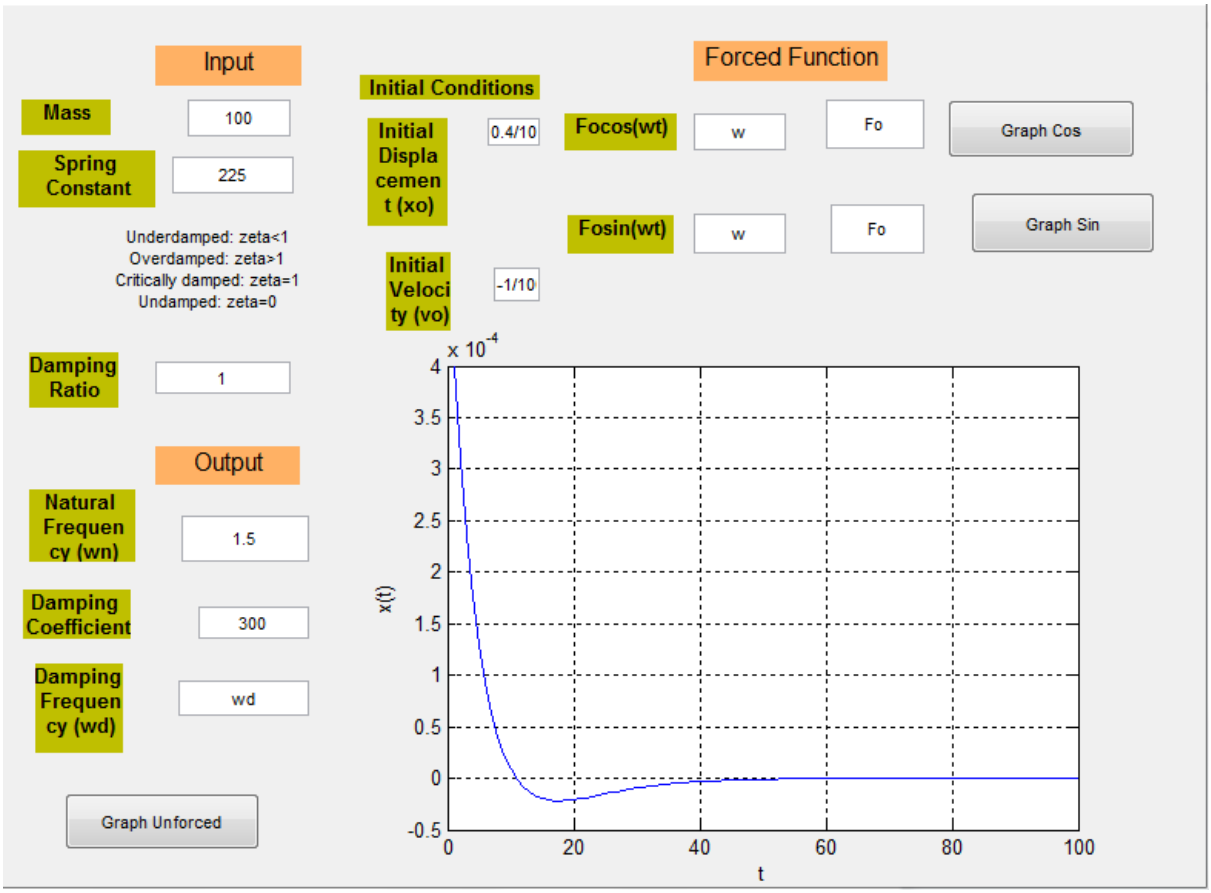


Fig.15: Critically Damped Display Example 1

The code and values used to create the graph above can be found in Appendix E. The experimental graph in Fig. 15 can be compared to the theoretical graph in Fig. 7 to prove that the GUI code actually works because it gives the same answer. The graph above also shows how a critically damped system only damps once and then stops which makes it different from an underdamped system which oscillates a number of times before stopping.

C. Forced Functions

a) Cosine

When the forced cosine function is undamped (c=0) it can be calculated using the total response in Eq.22 with the constants of integration calculated by Eq.23, 24 and 25. The inputs are placed in the same place as for the unforced systems, but for the cosine function the magnitude of the force and the frequency need to be inserted along the cosine line. In addition, the graph cos push button should be used so the right code is associated with it. The following Fig. 16 depicts the input and the graph for the total response versus time.

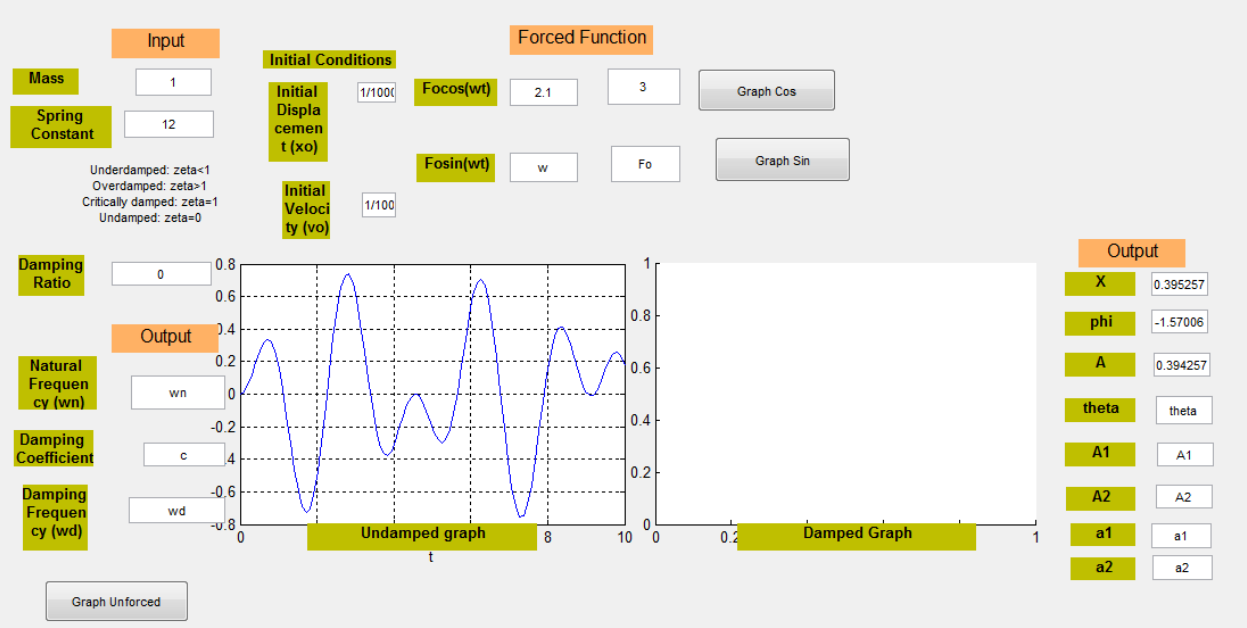


Fig.16: Display of Undamped Cosine Function Question 2.1

The case for an undamped forced function is the same as the case for an undamped unforced function, they both oscillate to infinity. The question for Fig. 16 was taken from the book Engineering Vibrations and can be found in Appendix F along with the code for the GUI.

For the damped case of a cosine forced function we only take into account the underdamped case ($\zeta<1$). The total response in this case is depicted by Eq. 27 and integration constants are calculated by the Eq. 28,29,30 and 31. Fig. 17 shows the total response vs. time graph, the inputs and outputs.

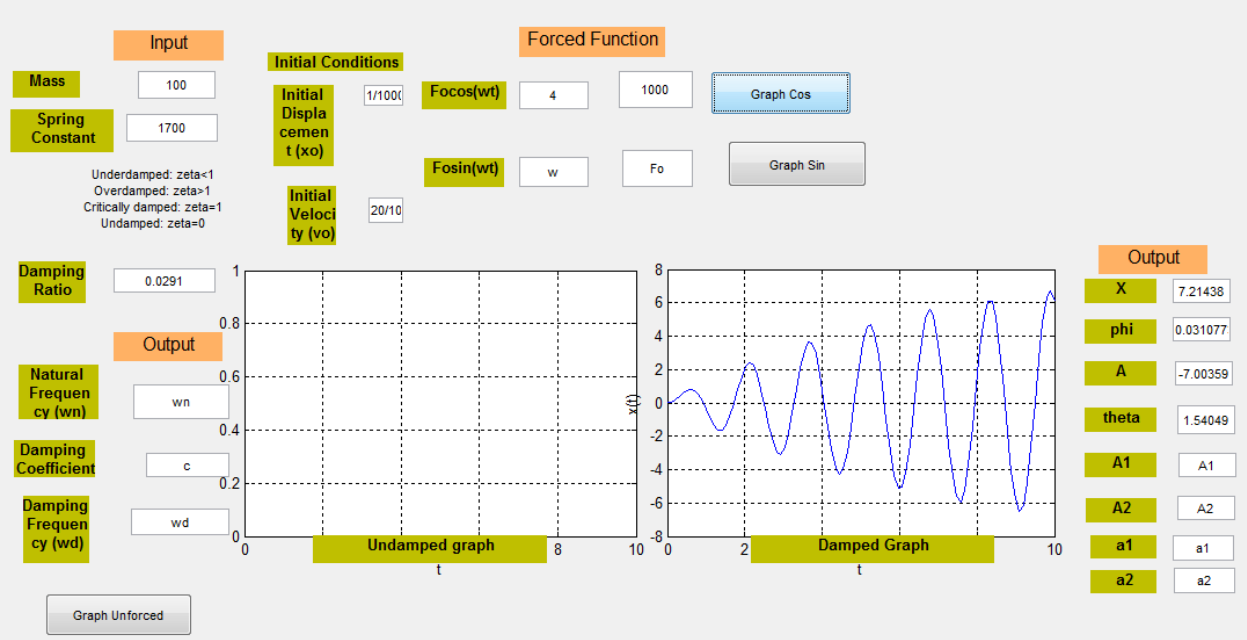


Fig.17: Display of Damped Cosine Forced Function Question 2.26

In this case the amplitude of the graph is increasing over time because the force is increasing the velocity of the mass spring system by 20mm/sec. As seen from Fig. 16 the undamped system doesn't have an exact increase in the amplitude compared to the Fig. 17 for a damped system which increases constantly by a certain number. Fig. 17 was created from question 2.26 from the Engineering Vibrations book, the detailed question along with the code can be found in Appendix F.

b) Sine

For the sine forced function the undamped (c=0) case is calculated by Eq. 33 which is the total response and the integrated constants are calculated by Eq. 34 and 35. In this case the frequency and the force need to be inserted in the sine line so the right code will be used and the Graph Sin push button should be used to graph it. Fig. 18 displays the graph, inputs and outputs for an undamped forced function.

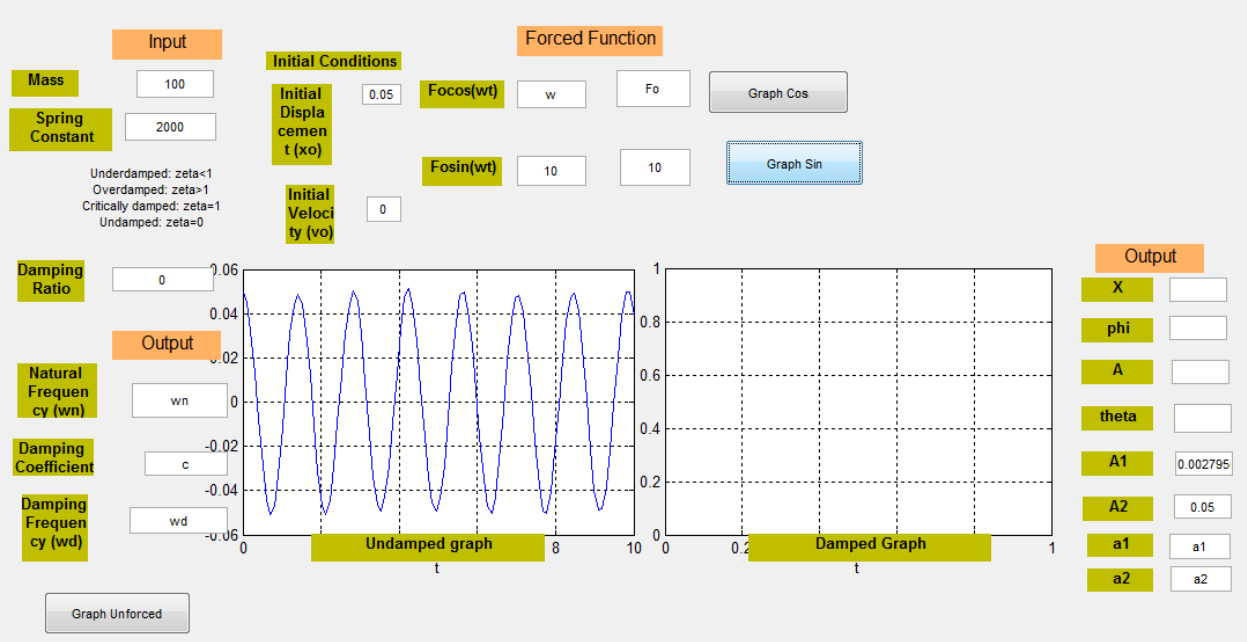


Fig. 18: Display of Undamped Sine Forced Function Question 2.8

The figure above was created using question 2.8 from the Engineering Vibrations textbook, the detailed question and code can be found in Appendix G. Comparing Fig. 18 with Fig.16, the undamped cosine function graph starts at 0 while the undamped sine function graph starts at around 0.05 meters which further proves that sine lags the cosine function by $\pi/2$.

The damped sine forced function follows the same principal as the damped cosine function in that it's going to be underdamped. To construct the total response for the damped sine forced function the concept above was used. The total response for the sine damped forced function is depicted by Eq. 36 and the integrated constants can be calculated using the same equations as for the damped cosine forced function. The total response vs time graph is displayed in Fig. 19 along with the inputs and outputs.

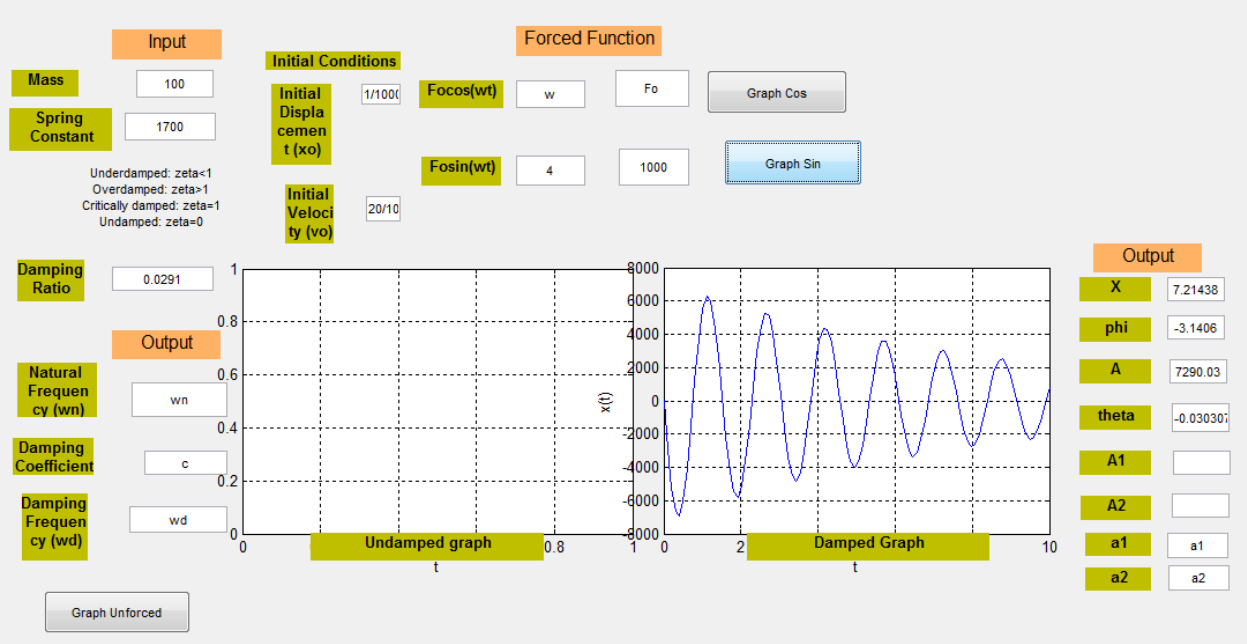


Fig. 19: Display of Damped Sine Forced Function

The same question as the one used to create the display for the damped cosine forced function was used to create Fig. 19, the only difference was that cosine was changed into sine. The detailed code for the figure above can be found in Appendix G. When Fig. 19 is compared to Fig. 17 we can see that Fig. 19 instead of increasing the amplitude over time it decreases which is the total opposite of the Fig. 17 which depicts the damped cosine forced function.

V. CONCLUSION

- Undamped systems oscillate to infinity
- Underdamped system has a time constant when it stops oscillating
- Overdamped and critically damped systems oscillate at most once
- The rule of sine lagging the cosine function can be applied to the undamped and damped forced function cases with a mass spring system
- The damped forced function system has a larger amplitude than an undamped forced function system
- The GUI display can help students visualize the difference between the undamped and damped system as well as the forced and unforced cases

VI. FUTURE RESEARCH

The code developed above in the GUI can be furthered implemented in the damped and undamped spring system teaching aid was created by students in the MED class. This device consists of two separate spring mass systems that have the same base which is excited by a rotating cam connected to a shaft and a motor (See Fig. 20) [3].

Fig. 20: Mass Spring System Teaching Aid [3]

The code above will need to be specifically designed to excite the base through the motor. Right now the base is excited manually, but by implementing this code the base can be excited automatically through a program. Then the structure featured in Fig. will be displayed in a glass case at the entrance of the Mechanical Engineering building. This will serve as a way for prospective students taking a tour of the building to interact with the damped and undamped system. The GUI display screen will be displayed in the glass case beside the mechanism and the students will be able to choose their input and the graph of the total response will be displayed alongside the mechanism moving to show the damping.

Another area of implementation is in a hardware devicee that can be used by current students in the Machine Dynamics and Control class. This hardware will contain the code above

and will help students check if their answers to the problems are right. It will also help them to see the difference between a damped and undamped system. This hardware will be able to be transported and connected to different computers so various students can use it. The most common hardware to use is the raspberry pi.

VII. REFERENCES

[1] Inman, J. Daniel. *Engineering Vibrations.* Pearson Education Inc. 4th edition. 2014.

[2] Wejinya, Uche. *Lecture Slides 3,6, and 7.* MEEG 3113 Machine Dynamics and Control. September 2013.

[3] Goll, Ted, Colton Stanton, Clint Paul, and John Vaughan. *Project 1 Report*. Rep. Spring 2015. University of Arkansas. Print.

[4] "Damping." *Cyberphysics*. 1 Jan. 2014. Web. 23 Mar. 2015. <http://www.cyberphysics.co.uk/topics/shm/damping.htm>.

VIII. APPENDICES

APPENDIX A: Underdamped System Question 1.41 Engineering Vibrations Using Matlab

```
m=1;
k=10;
c=2;
vo=1/1000;
xo=1/1000;
t=linspace(0,4,100);
wn=sqrt(k/m)
zeta=c/(2*m*wn)
wd=wn*sqrt(1-zeta^2)
A=(1/wd)*(sqrt(((vo+zeta*wn*xo)^2)+((xo*wd)^2)));
ang=atan((xo*wd)/(vo+zeta*wn*xo));
xt=A*sin(t*wd+ang).*exp(-zeta*t*wn);
display(c)
plot(t,xt)
grid on
ylabel('x(t)')
xlabel ('t')
```

Fig. A.1: Matlab Code to Generate Graph

```
>> underdamped141

wn =

    3.1623

zeta =

    0.3162

wd =

     3

c =

     2
```

Fig. A.2: Values Displayed in Command Window Matlab

APPENDIX B: Overdamped System 1.43 Using Matlab

```
m=100; %kg
k=225; %N/m
c=600; %kg/s
vo=-1/1000; %m/s
xo=0.4/1000; %m
t=linspace(0,15,100);
wn=sqrt(k/m) %natural frequency
zeta=c/(2*m*wn)  %damping ratio
a1=(-vo+((-zeta+sqrt((zeta^2)-1))*wn*xo))/(2*wn*sqrt((zeta^2)-
1));
a2=(vo+((zeta+sqrt((zeta^2)-1)*wn*xo)))/(2*wn*sqrt((zeta^2)-1));
xt=(exp(-zeta.*wn.*t)).*((a1.*exp(-wn.*(sqrt((zeta^2)-
1)).*t))+a2.*(exp(wn.*(sqrt((zeta^2)-1)).*t)));%general response
display(c)
plot(t,xt)
grid on
ylabel('x(t)')
xlabel ('t')
```

Fig. B1: Matlab Code for 1.43

```
>> overdamped143

wn =

    1.5000


zeta =

     2


c =

   600
```

Fig. B2: Values Displayed in Command window Matlab

APPENDIX C: Critically Damped System Example 1 Using Matlab

```
m=100; %kg
k=225; %N/m
c=300; %kg/s
vo=-1/1000; %m/s
xo=0.4/1000; %m
t=linspace(0,4,100);
wn=sqrt(k/m) %natural frequency
zeta=c/(2*m*wn)  %damping ratio
a1=xo;
a2=vo+wn*xo;
xt=(a1+a2.*t).*(exp(-wn.*t));%general response
display(c)
plot(t,xt)
grid on
ylabel('x(t)')
xlabel ('t')
```

Fig. C1: Matlab Code for Example 1 Critically Damped

```
>> criticallydampedexample1

wn =

    1.5000


zeta =

     1


c =

   300
```

Fig. C2: Values Displayed in Command Window Matlab

APPENDIX D: Undamped System Code and Example

```
A=(sqrt(((wn^2)*(xo^2))+(vo^2)))/wn;
    set(handles.A,'String',A)
    phi=atan((wn*xo)/vo);
    set(handles.phi,'String',phi)
    xt=A.*sin(wn.*t+phi);
    axes(handles.axes2)
    plot(xt)
    grid on
    ylabel('x(t)')
    xlabel ('t')
```

Fig. D1: Code for Undamped System

Question 1.19 [1]:

Plot the solution given by equation (1.10) for the case k=1000N/m and m=10kg for two complete periods for each of the following sets of initial conditions. a) xo= 0m, vo=1 m/s, b) xo=0.01m, vo= 0 m/s, and c) xo=0.01m, vo=1 m/s.

APPENDIX E: Damped System Code and Example

Underdamped

```
elseif zeta<1
    wd=wn*(sqrt(1-(zeta^2)));
    set(handles.damping_frequency,'String',wd)
    A=sqrt((((vo+zeta*wn*xo)^2)+((xo*wd)^2))/(wd^2));
    set(handles.A,'String',A)
    phi=atan((xo*wd)/(vo+zeta*wn*xo));
    set(handles.phi,'String',phi)
    xtud=A.*exp(-zeta*wn.*t).*sin(wd.*t+phi);
    axes(handles.axes1)
    plot(xtud)
    grid on
    ylabel('x(t)')
    xlabel ('t')
```

Fig. E1: Underdamped Code inside GUI

Question 1.41 [1]:

Consider a spring mass damper system, like the one Figure 19, with the following values m=1kg, c=2N/s and k=10N/m. a) Is the system overdamped, underdamped or critically damped? b) Compute the solution if the system is given initial coniditons xo=1 mm and vo=1mm.

Overdamped

```
else
    a1=(-vo+((-zeta+sqrt((zeta^2)-
1))*wn*xo))/(2*wn*sqrt((zeta^2)-1));
    set(handles.a_1,'String',a1)
    a2=(vo+((zeta+sqrt((zeta^2)-1)*wn*xo)))/(2*wn*sqrt((zeta^2)-
1));
    set(handles.a_2,'String',a2)
    xtov=(exp(-zeta.*wn.*t)).*((a1.*exp(-wn.*(sqrt((zeta^2)-
1)).*t))+a2.*(exp(wn.*(sqrt((zeta^2)-1)).*t)));
    axes(handles.axes1)
    plot(xtov)
    grid on
    ylabel('x(t)')
    xlabel ('t')
```

Fig. E3: Overdamped Code inside GUI

Question 1.43 [1]:

Consider the system $\ddot{x} + 4\dot{x} + x = 0$ for xo=1mm and vo=0mm/s. Is this system overdamped, underdamped or critically damped? Compute the solution and determine which root dominates as time goes on (that is, one root will die out quickly and the other will persist).

Critically damped

```
if zeta==1
    a1=xo;
    set(handles.a_1,'String',a1)
    a2=vo+wn*xo;
    set(handles.a_2,'String',a2)
    xtcd=(a1+a2.*t).*(exp(-wn.*t))
    axes(handles.axes1)
    plot(xtcd)
    grid on
    ylabel('x(t)')
    xlabel ('t')
```

Fig. E5: Critically Damped Code inside GUI

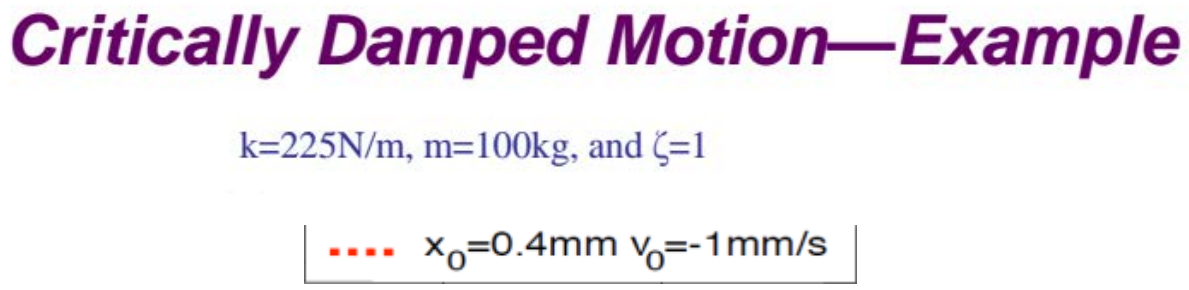


Fig. E6: Critically Damped Example 1 [2]

APPENDIX F: Forced System Code and Example

Cosine

```
if c==0
X=fo/((wn^2)-(w^2));
set(handles.X,'String',X)
phi=atan((wn*(xo-X))/vo);
set(handles.phi,'String',phi)
A=sqrt(((vo/wn)^2)+((xo-X)^2));
set(handles.A,'String',A)
xt=A*sin(wn.*t+phi)+X*cos(w.*t);  %units meter
axes(handles.axes1)
plot(t,xt)
grid on
ylabel('x(t)')
xlabel ('t')
```

Fig. F1: Code for Undamped Cosine Function

Question 2.1 [1]:

The forced response of a single degree of freedom, spring mass system is modeled by (assume the units are Newtons)

$$3\ddot{x}(t) + 12x(t) = 3coswt$$

Compute the magnitude of the forced response for the two cases w=2.1 rad/s and w=2.5 rad/sec/ Comment on why one value is larger than the other.

```
else  %underdamped zeta<1
theta=atan((2*wn*w)/((wn^2)-(w^2)));
set(handles.theta,'String',theta)
X=fo/(sqrt((((wn^2)-(w^2))^2)+((2*wn*zeta*w)^2)));
set(handles.X,'String',X)
phi=atan((wd*(xo-X*cos(theta)))/(vo+((xo-
X*cos(theta))*zeta*wn)-(w*X*sin(theta))));
set(handles.phi,'String',phi)
A=(xo-X*cos(theta))/sin(phi);
set(handles.A,'String',A)
xt=A.*exp(-zeta.*wn.*t).*sin(wd.*t+phi)+X.*cos(w.*t-theta);
axes(handles.axes2)
plot(t,xt)
grid on
ylabel('x(t)')
xlabel ('t')
```

Fig. F3: Code for Damped Cosine Function

Question 2.26 [1]:

A damped spring mass system modeled by (units are Newtons)

$$100\ddot{x}(t) + 10\dot{x}(t) + 1700x(t) = 1000cos4t$$

Is also subject to initial conditions xo=1mm and vo=20mm/s. Compute the total response x(t) of the system.

Sine

```
if c==0
A1=(vo/wn)-((w*fo)/(wn*((wn^2)-(w^2))));
set(handles.A1,'String',A1)
A2=xo;
set(handles.A2,'String',A2)
xt=A1*sin(wn.*t)+A2*cos(wn.*t)+((fo*sin(w.*t))/((wn^2)-(w^2)));
axes(handles.axes1)
 plot(t,xt)
 grid on
ylabel('x(t)')
xlabel ('t')
```

Fig. F5: Code for Undamped Sine Function

Question 2.8 [1]:

Consider the system in Figure P2.8, write the equation of motion and calculate the response assuming (a) that the system is initially at rest and (b) that the system has an initial displacement

```
else
theta=(atan((2*wn*w)/((wn^2)-(w^2))))-(pi/2);
set(handles.theta,'String',theta)
X=fo/(sqrt((((wn^2)-(w^2))^2)+((2*wn*zeta*w)^2)));
set(handles.X,'String',X)
phi=(atan((wd*(xo-X*cos(theta)))/(vo+((xo-X*cos(theta))*zeta*wn)-
(w*X*sin(theta)))))-(pi/2);
set(handles.phi,'String',phi)
A=(xo-X*cos(theta))/sin(phi);
set(handles.A,'String',A)
xt=A.*exp(-zeta.*wn.*t).*sin(wd.*t+phi)+X.*cos(w.*t-theta);
axes(handles.axes2)
set(handles.A2,'String','')
set(handles.A1,'String','')
plot(t,xt)
grid on
ylabel('x(t)')
xlabel ('t')
end
```

Fig. F7: Code for Damped Sine Function