NORTHEASTERN UNIVERSITY
Graduate School of Engineering

Thesis Title: Passive/Active Vibration Control of Flexible Structures

Author: Matthew Jamula

Department: Mechanical and Industrial Engineering

Approved for Thesis Requirement of the Master of Science Degree

______________________________________________    __________________
Thesis Adviser, Dr. Nader Jalili     Date

______________________________________________    __________________
Thesis Reader, Dr. Rifat Sipahi     Date

______________________________________________    __________________
Thesis Reader, Dr. Bahram Shafai     Date

______________________________________________    __________________
Department Chair, Dr. Jacqueline Isaacs    Date

Graduate School Notified of Acceptance:

______________________________________________    __________________
Dr. Sara Wadia-Fascetti,     Date
Associate Dean of the Graduate School of Engineering
PASSIVE/ACTIVE VIBRATION CONTROL OF FLEXIBLE STRUCTURES

A Thesis Presented

by

Matthew Thomas Jamula

to

The Department of Mechanical and Industrial Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

in the field of

Mechanical Engineering

Northeastern University
Boston, Massachusetts

April 2012
Abstract

More advanced technology and materials in industry lead to the implementation of lightweight components in order for miniaturization and efficiency. Lightweight components and certain materials however, are susceptible to vibrations. The flexible structures that make up these systems pose a great problem for vibration control. The detailed modeling of such systems greatly reduces the complexity of the control law. It is this reason that an analysis of the model as a continuous system was done. The distributed-parameters system was then effectively reduced to an equivalent lumped-parameters model. The use of this discrete system was the basis for controller design of these flexible structures.

However, even the best model of a system is not able to overcome the need of an advanced controller for vibration suppression. Flexible structures, which are a common problem in robotics, represent nonlinear terms such as damping. Piezoelectric actuators or transmissions using gears can often be subject to nonlinear effects such as hysteresis or backlash. Even a multi-part system could be subject to frictions or other conditions that could be found at boundary conditions of individual pieces. Thus, a controller is proposed that will account for unmodeled dynamics of the system.

This controller will also have the ability to reject external disturbances while accounting for varying parameters. It is rare that the properties of a structure do not change over time or with environmental factors such as temperature or humidity. Therefore, the controller must be able to account for these changes whether the change comes within the materials or in the joints of a structure.

A robust adaptive controller with perturbation estimation will guarantee stability for all of the noted effects. It will be robust enough to account for completely unmodeled dynamics while rejecting unknown disturbances. The adaptive law within the controller will be an on-line estimation of the parameters modeled within the system. The simulations of this advanced controller show the stability of the system and prove its robust and adaptive features when subject to varying
internal or external conditions and disturbances. A proposed experimental setup is also discussed.
Acknowledgements

First, I would like to give thanks to Draper Laboratory who funded and supported my studies and research at Northeastern University.

I’d especially like to thank my advisor Dr. Nader Jalili for his patience and support through my Master’s program. His guidance helped shape my knowledge gained through studies and research.

Many thanks go to the faculty and staff within the department of Mechanical Engineering. Their knowledge of the technical aspects of my studies made my research a much more feasible task.

Finally, I’d like to thank all my family and friends for their love, support and encouragement.
# Table of Contents

1. **Introduction** .......................................................... 9
   
   1.1. **Thesis Outline** ........................................... 13

2. **Theory** ................................................................. 14
   
   2.1. **Vibrations** .................................................. 14
       
       2.1.1. **Discrete Systems** ................................. 15
       
       2.1.2. **Continuous Systems** ....................... 16
   
   2.2. **Vibration Control** ..................................... 24
       
       2.2.1. **Passive Control** ................................. 25
       
       2.2.2. **Active Linear Control** ..................... 26
       
       2.2.3. **Advanced Control** ........................... 27

3. **Data & Analysis** .................................................. 31
   
   3.1. **Simulation** .................................................. 31
       
       3.1.1. **Model Derivation** ............................... 31
           
           3.1.1.1. ANSYS – MATLAB Interface ............... 35
           
           3.1.2. **Passive Suspension Models** ............. 38
           
           3.1.3. **Active Linear Controller Design** ........ 42
               
               3.1.3.1. PID Controller Design ..................... 42
               
               3.1.3.2. LQR Design .................................... 45
           
           3.1.4. **Robust Adaptive Controller** ............. 49
               
               3.1.4.1. Advanced System Model ................. 57
   
   3.2. **Experimentation** ........................................ 60
       
       3.2.1. **Part Design and Assembly** ................. 60
       
       3.2.2. **Experimental Setup** .......................... 62

4. **Conclusion** .......................................................... 64
   
   4.1. **Summary and Discussion** ............................ 64
   
   4.2. **Future Work** ............................................ 66

5. **Appendices** .......................................................... 67
   
   5.1. **Appendix A** – Equivalent Parameters Calculations (MATLAB and Maple) .................................................. 67
   
   5.2. **Appendix B** – Passive Simulink Diagrams ............ 71
   
   5.3. **Appendix C** – Active Suspension Models .......... 73
   
   5.4. **Appendix D** – Controller Tests, MATLAB m-files 74

6. **References** .......................................................... 83
Table of Figures

Figure 1 - Continuous versus discrete systems [16] .......................................................... 14
Figure 2 - Stresses on each face of a deformable body [16] ............................................. 16
Figure 3 - Strain-displacement relations [16] ................................................................. 17
Figure 4 - Forces acting on a taut string [17] ............................................................... 19
Figure 5 - Relating deformed position of string to length of differential element dx [17] .................................................................................................................. 20
Figure 6 - Thin plate in transverse vibration [16] .......................................................... 23
Figure 7 - Variable Control Structures [20] ................................................................. 28
Figure 8 - Sliding Hyperplane [20] ............................................................................... 28
Figure 9 - Chattering in SMC [21] ............................................................................. 29
Figure 10 - Adaptive Control [21] ............................................................................... 30
Figure 11 - Equivalent Model of Clamped-Free-Clamped-Free Plate ..................... 32
Figure 12 - Equivalent Lumped-Parameters Model .................................................. 34
Figure 13 - Nonlinear Effects on Continuous System ............................................. 34
Figure 14 - ANSYS Geometry .................................................................................. 35
Figure 15 - Continuous ANSYS Model ................................................................. 36
Figure 16 - Passive Suspension Systems ................................................................. 38
Figure 17 - Continuous Mass, Spring, Damper System ........................................ 38
Figure 18 - Suspension Model 1 Comparison .......................................................... 39
Figure 19 - Suspension Model 2 Comparison ............................................................ 40
Figure 20 - Suspension Model 3 Comparison ............................................................ 40
Figure 21 - Suspension Models ............................................................................... 41
Figure 22 - Suspension Variation Comparison ....................................................... 41
Figure 23 - System with PID Controller ................................................................. 43
Figure 24 - PID Gain Test ....................................................................................... 44
Figure 25 - System Response, PID (blue) vs Simple Feedback (red) ................. 45
Figure 26 - System with LQR Controller ................................................................. 46
Figure 27 - LQR Gain Test ..................................................................................... 47
Figure 28 - LQR Set Point Tracking ....................................................................... 47
Table of Tables

Table 1 - PID Gain Tuning ................................................................. 44
Table 2 - Adaptability of Robust Adaptive Controller .................. 56
Table 3 - Robustness of Robust Adaptive Controller ................. 56

List of Symbols

\( \lambda \) controller gain proportional to the error in the system (L1, L2, L3)
\( \alpha \) controller gain in robust adaptive controllers
\( \beta \) controller gain in robust adaptive controllers
\( A \) amplitude of narrowband disturbance
\( \omega \) frequency of narrowband disturbance
\( \gamma \) weighting values in adaptation law
\( \varphi_{\text{est}} \) perturbation estimation
\( x_d \) desired trajectory
1 Introduction

As the field of materials and technology advances, the engineering industry is coming up with light-weight, smaller, and more cost-effective products. The drive to achieve more compact and inexpensive structures leads to systems that have more flexibility and present a tough control problem. The vibration problem can be caused by external disturbances, internal uncertainties such as frictions or ignored higher order dynamics. These issues are seen all over the field of mechanics whether it is robotic systems, spacecraft structures, or optical systems. Robotic systems are comprised of flexible links with variations of loading. Spacecraft structures often undergo disturbances and vibrations from the physical environment. The problem of eliminating vibrations is especially valuable with the use of optics due to the easy distortion of an optical surface.

This thesis explores the need for vibration control of flexible structures. A flexible beam or plate must be made able to track a desired trajectory or motion while rejecting the effects of uncertainties in the model as well as external disturbances from the environment. The use of nonlinear control allows motion tracking while reducing sensitivity to the plant parameters and guaranteeing stability of the system. Nonlinear control is very effective because it can also be used for linear systems, whereas linear control cannot be used effectively for nonlinear systems. Sliding mode control is a form of variable structure control, where the nonlinear plant is controlled by a switching mechanism that achieves stability of the system by the use of two structures rather than one, represented by a sliding hyperplane. There have been many efforts to control flexible beams and plates in the past, but some efforts have assumed that the parameters of the system are known or made the use of disturbance observers for robust control. By using perturbation estimation, the sliding mode control design will allow for online adaptation to the disturbances therefore becoming insensitive to modeling uncertainties.
Two common techniques useful to vibration control are by vibration isolation and vibration absorption. Vibration isolation eliminates the vibrations at the point of attachment to the disturbance [1]. For a system that is mounted at one point this may be very effective. However, if there are different forces at multiple points on the object of interest, this may lead to complicated controls. Vibration isolation is successfully shown by Huang where four actuators are paired with passive supports to eliminate the vibrations acting on flexible equipment [2]. In this study, there is a uniform disturbance on the flexible equipment by a rigid base. Although this base is also examined as flexible and there are four points of disturbance, it is an equal disturbance at each point.

Another method to eliminate vibrations is by using a vibration absorber. Vibration absorption consists of a secondary system that is added to the existing model [1]. This extra system usually consists of a mass, spring, and damper. By actuating the additional mass, the energy from the plant will be dissipated through the inertial actuator, thus stabilizing the system. In a study performed by Wu, it is seen that a spring-mass-damper absorber located at the end of a flexible plate can successfully eliminate vibrations of the structure for tuned modes [3]. In other cases, a piezoelectric patch can be applied over some beams and plates to actuate the system over a distributed area [4]. As seen in Kumar’s study, a linear quadratic regulator (LQR) was implemented with a piezoelectric patch to absorb the vibrations of the beam. Kumar went on to explore having up to five piezoelectric sensor-actuator combinations placed in optimal locations on the beam. This allowed for vibration absorption of multiple modes.

There have been many efforts to control flexible structures in the past, but some efforts have used passive or active linear control methods which assumed that the parameters of the system are known [6,7,8]. In studies where passive vibration control or active linear vibration control has been explored, the most important step is to develop a very detailed model of the system. Quan explored the use of a switching vibration absorber to control a multiple degree of freedom structure [6]. Rahman successfully used a piezoelectric actuator on a thin vibrating plate to
dissipate the energy from the system through the use of a proportional controller [7]. This reduced the energy, but it did not completely suppress the vibrations. Sethi used a linear quadratic regulator and the use of a state observer to eliminate the vibrations to a frame structure using piezoceramic sensors and actuators [8]. In each of these cases, vibrations and energy of the structures were reduced.

However, this assumed that all of the parameters in the system were known, and that the model was not changing. In these three cases, a detailed model of the system was derived and assumed constant over time. This is almost never the case, as parameters can vary over time or with changes in the environment. Adaptive controllers on the other hand, need very little or even no initial information about the plant parameters, but because of the online estimation, the parameters will be learned and tracked as they vary. Another reason that the linear controllers are not adequate for controlling vibrations is that they don’t account for unmodeled dynamics or nonlinear terms within the model. This is where a robust controller comes into play – it can deal with these unmodeled dynamics in addition to quickly varying parameters and external disturbances [9].

Some attempts to control uncertain parameters have been done with the use of adaptive controllers. One study done by Xian uses an online estimation technique to learn the parameters of the system [10]. This parameter adaptation is used in the disturbance estimation and fed back into the controller to account for a varying system. However, in this study the model of the system was assumed to be linear. This is not always the case as many systems contain flexible structures that have damping terms associated with them. Also, if the system is assembled with motors or even just clamped with screws and bolts, it could still suffer from nonlinearities such as backlash or stick-slip joint friction. Therefore, it is necessary to come up with a controller that has the robustness to guarantee stability regardless of any nonlinearities in the system.

Yet another few studies show the benefits of having a robust adaptive controller to account for completely uncertain parameters and nonlinear dynamics within the system. Liu explores using a robust adaptive controller to stabilize a system in
the presence of nonlinear and unmodelled dynamics [11]. In the sliding mode controller, Liu implemented a nonlinear damping term which accounts for the unmodelled dynamics and bounded disturbances of the system. This guaranteed stability of the system for bounded disturbances and improved tracking performance by correctly choosing the design parameters. In a similar study, Hu achieves attitude tracking control as well as suppressing undesired disturbances on the structure [12]. Hu uses a sliding mode controller that includes a nonlinear switching function that was able to account for uncertain parameters and dynamics of the system such as nonlinearities that would not even be modeled. The study also suppressed these undesired vibrations assuming that that the disturbance was bounded. Although actively vibration control was successfully achieved, the addition of perturbation estimation to the control law would allow for rejection of any unknown disturbance. In an environment such as spaceflight, this ability to account for completely unknown disturbance would be a significant improvement over the current design.

Elmali and Olgac took the sliding mode controller a step further when they implemented a sliding mode controller with perturbation estimation (SMCPE) [13]. This study proved that even in the absence of an adaptation law, the controller could successfully account for unmodeled dynamics and reject external disturbances without needed upper bounds on the perturbations. Also, the tracking was improved with the nonlinear controller.

In order to combine all the previously mentioned controller designs and incorporate them into one leads to a robust adaptive controller that includes perturbation estimation. Ghafarirad et al. included all these features in a controller as well as implementing a state observer to estimate the immeasurable states of the system [14,15]. In each of these studies, position was assumed to be measurable, and a state observer was created for the velocity and acceleration states. These were used for the basis of a sliding mode control scheme that accounted for system dynamics and nonlinearities that are common in piezoelectrics such as hysteresis and backlash. In addition to developing an
observer-based sliding mode controller to have a robust feature of the controller, Ghafarirad proved guaranteed stability using the Lyapunov method and thus created an adaptation law. This adaptive law was used to feed estimated system parameters back into the controller as well as the perturbation estimation. By using the controller to learn the environmental forces or disturbances to the system, stability could be guaranteed despite uncertain parameters, unmodeled nonlinear dynamics and unknown disturbances that weren’t assumed bounded. This controller design achieved disturbance rejection and system adaptation while precisely tracking desired trajectories.

In this thesis, a robust adaptive control with perturbation estimation is derived and simulated for a flexible structure. The structure was first modeled as a thin plate before expanding to a more complex geometry. The adaptation law was derived and proved effective as well as the sliding mode controller which allows the guaranteed stability of the system regardless of uncertain parameters and unmodeled dynamics.

1.1. Thesis Outline

This thesis is organized as follows: Chapter 2 is a background of the theory split into two sections. The first section is the theory behind the vibrations of flexible structures. Vibrations in deformable bodies and taut strings are reviewed before moving on to calculations involving a thin plate. The second section consists of a discussion of theory and design for vibration control. This covers passive control before exploring linear and nonlinear forms of active vibration control. Chapter 3 concerns the modeling and simulation of the designed controllers. It discusses the link between discrete and continuous modeling of the system and how it is implemented in the controller. It also goes over the implementation of active controllers including a robust adaptive controller highlighting the gain tuning guidelines – controller effort and performance. Chapter 4 provides conclusions of the simulations and calculations and also gives a review of the experimental setup and future work to be conducted.
2 Theory

The content of this thesis is split into two main components: the model of the system, and the control of the system. In order to derive the best model of the system, the vibrations of flexible structures using both lumped-parameters and distributed-parameters methods is explored. For vibration control, numerous methods are looked into including passive simple feedback control, active linear control, and advanced robust adaptive control.

2.1 Vibrations

Vibrating systems can be categorized by two types of models – discrete and continuous. In real-world applications, almost everything is a continuous, or distributed-parameters, system. However, there comes a point when modeling a discrete, or lumped-parameter system is a more efficient means of modeling for control. Discrete systems are models in which a mass can be defined as a rigid body that does not deform, and the physical components of the system can connect to the mass at a specific point. Essentially, if the properties of the system can be “isolated,” it can be modeled as a discrete system [16]. For example if a beam can be represented by a point mass, it is considered a lumped-parameters system. On the contrary, if the beam is non-uniform, or if there is a distributed loading on the beam, then the physical parameters cannot be lumped together and it is best modeled as a distributed-parameters system.

![Figure 1 - Continuous versus discrete systems [16]](image)
2.1.1 Discrete Systems

The generic form of a single degree of freedom (SDOF) lumped-parameters model is represented by the equation:

\[ m\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t) \]  \hspace{1cm} (1)

This can be rewritten to better visualize the vibrations in the system. The model can also be expressed as:

\[ \ddot{x}(t) + 2\zeta\omega_n\dot{x}(t) + \omega_n^2x(t) = \frac{1}{m}f(t) \]  \hspace{1cm} (2)

where the damping ratio, \( \zeta = \frac{c}{2\sqrt{mk}} \), and the natural frequency, \( \omega_n = \sqrt{\frac{k}{m}} \). The lumped-parameters systems are written for systems with rigid masses.

If there is more than one body involved, the system is said to be a multiple degree of freedom (MDOF) system. In MDOF systems, the mass, stiffness, and damping parameters all form matrices. The form can remain the same as in equation (1).

\[ [M]\ddot{x}(t) + [C]\dot{x}(t) + [K]x(t) = \bar{f}(t) \]  \hspace{1cm} (3)

In order to find the natural frequencies and modeshapes of a system, it must be calculated from the free, undamped form. In MDOF systems specifically, this entails rewriting the equation to:

\[ [M]\ddot{x}(t) + [K]\ddot{x}(t) = 0 \]  \hspace{1cm} (4)

Just as in the SDOF approach, where \( \omega_n = \sqrt{\frac{k}{m}} \), one can find the eigenvalues of the system in terms of the mass and stiffness matrices.

\[ \Delta(\omega^2) = det[K - \omega^2M] = 0 \]  \hspace{1cm} (5)
2.1.2 Continuous Systems

If the mass is considered deformable or flexible or the physical components of the system such as spring and dampers are not massless, the system is continuous. In distributed-parameters systems, the equations of motion are written for segments of the system, while the reaction at the end of the model is defined as the boundary-value problem [17].

The first step in deriving the equations of motion is to determine all the forces acting on the deformable body. The figure below shows all the forces acting on an infinitesimal section of a continuous system.

![Figure 2 - Stresses on each face of a deformable body [16]](image-url)
From this figure, and where \( x_b, y_b, \) and \( z_b \), are body force per unit volume, one can determine the differential equations of equilibrium. After simplification, the three equations represent the forces per unit volume acting on any point within the volume.

\[
\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + x_b = 0 \quad (6)
\]

\[
\frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \tau_{yz}}{\partial z} + y_b = 0 \quad (7)
\]

\[
\frac{\partial \sigma_{zz}}{\partial z} + \frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} + z_b = 0 \quad (8)
\]

In addition to finding stresses of a deformable body, the strain-displacement relationships can be found. In Figure 3 \( X_i \) stands for the undeformed configuration while \( x_i \) stands for the deformed shape.

Letting \( dL \) be the distance \( AB \) in the undeformed configuration and \( dl \) be the distance \( AB \) in the deformed body, the measure of strain can be denoted by:

\[
ds = (dl)^2 - (dL)^2 \quad (9)
\]

After some manipulation and substitutions, the strain-displacement relations are the resulting equations below [16].
In order to implement the effects of the mechanics into the vibration problem, stress-strain relationships need to be developed. These relationships can be generalized as [16]:

\[ e_{xx} = S_{xx} = \frac{\partial u}{\partial x}, \quad e_{yy} = S_{yy} = \frac{\partial v}{\partial y}, \quad e_{zz} = S_{zz} = \frac{\partial w}{\partial z} \]  

(10)

\[ e_{xy} = S_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}, \quad e_{xz} = S_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}, \quad e_{yz} = S_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \]  

(11)

The derivation of the vibration problem for a distributed-parameters system can be done in multiple ways. Newtonian mechanics is one of these methods, but can be vary tedious. The extended Hamilton principle is another of these methods, which uses the Lagrange energy equation to derive the equation of motion and boundary conditions of the system. The Lagrange equation consists of the total potential energy of the system subtracted from the kinetic energy;

\[ L = T - V. \]

The extended Hamilton principle then sums the Lagrange and the work due to nonconservative forces and integrates over time. To show the method of the extended Hamilton principle using the Lagrange energy equation, the derivation of the string vibration problem is reviewed.
For the string problem the kinetic energy is defined by the dynamics of the system. In this case it will be in terms of the mass per unit length, \( \rho(x) \), and the velocity at which the string moves. The potential energy is defined using the basic stresses and strains as discussed above to represent the total internal energy of the system [16].

\[
T(t) = \frac{1}{2} \int_0^L \rho(x) \left[ \frac{\partial y(x,t)}{\partial t} \right]^2 dx \tag{13}
\]

\[
V(t) = \frac{1}{2} \int_0^L T(x)[ds - dx] \tag{14}
\]

The work of nonconservative forces can also be taken from the figure as it is seen that the distributed force \( f(x,t) \) induces the displacement \( y(x,t) \).

\[
\delta W_{nc} = \int_0^L f(x,t) \delta y(x,t) dx \tag{15}
\]
Since the string is assumed to be in tension, Figure 5 shows that the strain term, \( ds \), can be represented as a function of the length of a differential element \( dx \).

\[ \text{Figure 5 - Relating deformed position of string to length of differential element } dx \] [17]

Assuming that the slope of the deflected string is small, \( \frac{\partial y}{\partial x} \approx 0 \), the corresponding equation for strain is

\[ ds = \left[ 1 + \frac{1}{2} \left( \frac{\partial y}{\partial x} \right)^2 \right] dx \] (16)

which then rewriting the potential energy gives

\[ V(t) = \frac{1}{2} \int_0^L T(x) \left[ \frac{\partial y(x, t)}{\partial x} \right]^2 dx \] (17)

Writing the extended Hamilton principle,

\[ \int_{t_1}^{t_2} \left( \delta T - \delta V + \delta W_{nc} \right) dt = 0, \quad \delta y(x, t) = 0, \quad 0 \leq x \leq L, \quad t = t_1, t_2 \] (18)

Using mathematical operations such as integrating by parts, and grouping like terms together, the differential equation and corresponding boundary conditions were found to be:

\[ \frac{\partial}{\partial x} \left( T \frac{\partial y}{\partial x} \right) + f = \rho \frac{\partial^2 y}{\partial t^2}, \quad 0 < x < L \] \hspace{1cm} (19a)

\[ T \frac{\partial y}{\partial x} \bigg|_{x=0} = 0, \quad x = 0 \] \hspace{1cm} (19b)
Both of these boundary conditions are said to by geometric, or essential, because it can be written based only on geometric conditions [17]. Unconventional boundary conditions can exist if the end of the string was held by a spring or a dynamic component. This type of boundary condition is said to be a natural condition.

In addition to external forces contributing to the work of nonconservative forces, $\delta W_{nc}$, there is also internal damping that can be included. If there were no external forces but damping was involved, then the work would be defined as

$$\delta W_{nc} = \int_0^L f_c \delta u dx$$  \hspace{1cm} (20)

where, for viscous damping the term $f_c$ is defined as

$$f_c = c \dot{u}(x,t) = c \frac{\partial u(x,t)}{\partial x}$$  \hspace{1cm} (21a)

For other types of damping, $f_c$ is defined for structural damping and Kelvin-Voigt damping respectively, as:

$$f_c = B \frac{\partial u^2(x,t)}{\partial x \partial t}$$  \hspace{1cm} (21b)

$$f_c = D \frac{\partial u^5(x,t)}{\partial x^4 \partial t}$$  \hspace{1cm} (21c)

Going back to the differential equation found of a string, it is a function of a spatial variable $x$ and time variable $t$, and can therefore be split into a time and space equation [17].

$$y(x,t) = X(x)T(t)$$  \hspace{1cm} (22)

Rewriting the string equation of motion leads to the form,
From here, the method of separation of variables can be used to set each side of the equation equal to the same constant. Doing this and exploring the right side of the equation gives:

\[
\frac{1}{\rho(x)X(x)} \frac{d}{dx} \left[ T(x) \frac{dX(x)}{dx} \right] = \frac{1}{T(t)} \frac{d^2T(t)}{dt^2}, \quad 0 < x < L
\]  

(23)

This equation can be used to find the eigenfrequency equation and ultimately the natural frequencies of the system. Meanwhile, by plugging in the boundary conditions to the spatial side of the equation, one can get the general eigenfunction. Once the natural frequencies are known they can be substituted into the eigenfunctions to find the overall solution.

When the structure of interest in the vibration problem includes bending forces, such as a beam or a plate, the calculations get much more involved. Many of the assumptions made for classic plate theory, or Kirchhoff theory, are based on the assumptions from Euler-Bernoulli beam theory [18]. Some of these assumptions include that the thickness of the plate is small compared to the length and width, the effect of rotary inertia is neglected as is shear deformation, and also that transverse deflection is small compared to the thickness of the plate [18].
Another assumption made is that the plate is in a state of plane stress, so the stress-strain relationships in terms of the transverse displacement \(w(x,y,t)\) simplify to:

\[
\sigma_{xx} = \frac{E}{1 - \nu^2} (\varepsilon_{xx} + \nu \varepsilon_{yy}) = -\frac{Ez}{1 - \nu^2} \left( \frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \tag{25a}
\]

\[
\sigma_{yy} = \frac{E}{1 - \nu^2} (\varepsilon_{yy} + \nu \varepsilon_{xx}) = -\frac{Ez}{1 - \nu^2} \left( \frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \tag{25b}
\]

\[
\tau_{xy} = \frac{E}{2(1 + \nu)} \varepsilon_{xy} = -\frac{Ez}{1 + \nu} \frac{\partial^2 w}{\partial x \partial y} \tag{25c}
\]

Once the stress-strain relationships are determined, the internal work found from the strain energy of the system and the kinetic energy can be shown as:

\[
\delta U = \int \int \int_V \left[ \sigma_{xx} \delta \varepsilon_{xx} + \sigma_{yy} \delta \varepsilon_{yy} + \tau_{xy} \delta \varepsilon_{xy} \right] dV \tag{26}
\]

\[
T = \frac{1}{2} \int_0^a \int_0^b \rho(x,y) \left( \frac{\partial w(x,y,t)}{\partial t} \right)^2 dx dy \tag{27}
\]
The external force acting on the plate is represented in the work of nonconservative forces as shown below.

\[
\delta W^e = \int_0^a \int_0^b P(x,y,t) \delta w(x,y,t) dx dy
\]  

(28)

These three terms are then plugged into the Hamilton extended principle. After much manipulation, substitution, and integrating by parts, the governing equation of motion for transverse vibration of a plate is [16]:

\[
\rho t_b \frac{\partial^2 w(x,y,t)}{\partial t^2} + D \left( \frac{\partial^4 w(x,y,t)}{\partial x^4} + 2 \frac{\partial^4 w(x,y,t)}{\partial x^2 \partial y^2} + \frac{\partial^4 w(x,y,t)}{\partial y^4} \right) = P(x,y,t)
\]  

(29)

where D is the flexural rigidity of the plate, and is given as \(D = \frac{Et_b^3}{12(1-\nu^2)}\). With the use of the biharmonic operator \(\nabla^4\), the governing equation of motion can be simplified to:

\[
\rho t_b \frac{\partial^2 w(x,y,t)}{\partial t^2} + D \nabla^4 = P(x,y,t)
\]  

(30)

The boundary conditions can be seen by the results of the remaining terms once the equation of motion is formed. However, it can also be thought of mechanically. Below is a chart of some possible boundary conditions. If the plate is clamped, then it will have zero displacement as well as zero slope \(\frac{\partial w}{\partial x}\). However, since it is clamped, there will be a bending force produced as well as a shear force at the point of clamping. A cantilever plate, for example, will only have one clamped side as opposed to three sides that have displacements and slopes while the bending and shear forces do not exist.

### 2.2 Vibration Control

The detailed modeling of a system is just the first step in vibration control. Creating an accurate model makes for a less complicated controller design as well
as likely a more efficient performance of the controller. One option in vibration reduction is to determine whether to use vibration isolation or vibration absorption. One method could produce a different system architecture and controller design than the other. Vibration isolation consists of controlling the system at the point of attachment to the disturbance. For example, if a car is driving along a road, a vibration isolation system would be included in the suspension. However, if a vibration absorber was used, there would be a secondary system attached to the object of interest that would mimic the energy of the vibrations in order to transfer it to other components [16]. For the purposes of control design in this study, vibration isolation at the point of attachment was the method of choice.

Creating a good passive structure is important in controller design because if an active controller fails, the passive components will be there to help control the system. When exploring the realm of active control, linear controllers successfully work for a linear system with known parameters. However, when the model of the system becomes complex as a result of multiple masses, multiple modes, or unmodeled dynamics, the simple linear controllers are inadequate.

Herein lies the need for an advanced controller. The robust adaptive control theory has been around since the mid-1980’s and is a large part of the industry today. The idea of robust adaptive control allows a system to account for unknown or changing parameters of the system while at the same time rejecting disturbances. The use of a sliding mode controller with the adaptation techniques makes the controller robust to external disturbances and unmodeled dynamics and nonlinearities.

2.2.1 Passive Control

Before one designs an overly complex active controller, they should look into the affects the passive components have on the system. One method of doing this is to evaluate different combinations of passive components and compare the response of the system. It is seen that increasing the spring constant of the system
will reduce the amplitude of vibrations. However, unless a good balance with a damping component is found, the energy going into the system will not dissipate quickly and will instead vibrate for a significant length of time.

2.2.2 Active Linear Control

Once basic stability analysis can be done for the system with passive support structure, active control can be explored. Some of the most used active control involves linear controllers such as the proportional integral derivative controller (PID) or the linear quadratic regulator (LQR). The PID controller focuses on the use of classical design control, such as the use of root locus and frequency response of a transfer function. The LQR controller on the other hand, is a method of optimal control, or modern control design, used in the state-space domain [18]. The advantage of state-space control becomes more apparent with multiple input or multiple output systems. However, this study will focus more along the lines of the single-input-single-output (SISO) systems.

As stated above, the PID controller is mainly focused on the classical control design of systems. That is, linear systems for which the transfer function can be derived. PID control is very useful for transient response. The proportional feedback constant, P, is useful to manipulate the natural frequency of the system and therefore control the amplitude of vibrations. However, the higher the stiffness term becomes without any way to control the damping, the system could be nearly impossible to control. The addition of an integral constant, I, gives the adjustment needed for the damping, or energy dissipation of the system. Pairing the integral term along with the proportional constant, gives the controller a way to minimize steady state error, while having the ability to minimize the effects of disturbances to the system. Finally, the derivative constant, D, controls the speed or response of the controller.

These controllers are usually be designed by using stability analysis methods such as root locus or frequency response. In each controller designed, the tuning of the gains was done to obtain the best controller performance while taking into
consideration the effort needed for control of the system. The performance of the controller is a function of the error of the system, while the controller effort is a function of the control output to the system. In all physical systems, there is some limitation to the effort available, such as power to drive the controller, or frequency of actuation.

\[
\text{Controller Effort} = \int_{t_0}^{t_f} u^2(t)dt \tag{31}
\]

\[
\text{Controller Performance} = \int_{t_0}^{t_f} e^2(t)dt \tag{32}
\]

Despite its name, the LQR can be used to control both linear and nonlinear systems. For nonlinear systems, they have to be linearized in the state-space method to be controlled by the LQR. Although it is possible to control a nonlinear system with and LQR, it is not the optimal method since there is an iterative process needed to find the best weighting factors for certain design criteria. In this case, the design criteria consist of controller effort and performance.

As previously mentioned, the LQR is developed in the state-space. This can allow the controller to estimate unknown states. In the PID controller as well as the original LQR design, all states of the system are assumed accessible. However, this is not always the case. Whether there is a physical limitation such as not enough space to mount an accelerometer or another sensor, or if there are states that are simply not measureable such as third and fourth derivatives. In the state-space method, an estimator can be created that will allow the LQR to run as if all the states are known.

### 2.2.3 Advanced Control

The robust adaptive controller design involves many stages. The first of which is to choose the sliding hyperplane. In sliding mode control (SMC), a stable system can be obtained by switching between two unstable structures. This is a type of the variable control structure method.
The sliding hyperplane is a line defined in the state-space domain that is on the boundary of two different control structures. The SMC will switch back and forth between these two structures, often at high frequencies. The switching method will prevent the system from being controlled by just one structure. Instead a line will result from the switching. This line or plane is the sliding hyperplane mentioned above.

One common side effect with using sliding mode controllers is the chattering problem. This is most common when the signum switching function is used. This is concerning because often chattering can excite higher order dynamics of the system that were neglected during modeling [21].
To remedy this situation, other types of switching functions such as the saturation function or the tangent hyperbolic function are used. Choosing the sliding hyperplane to be proportional to the error of the system is a common method.

In this particular version of the sliding mode controller, it will also include perturbation estimation (SMCPE). In normal SMC, the disturbance is assumed to be bounded by an upperbound. With the perturbation estimation, the SMCPE allows for the estimation of the disturbance to become adaptive [22]. The perturbation estimation will use a similar model to that of the plant, but will also take into consideration the controller at the previous timestep.

After defining the hyperplane, the stability of the system is guaranteed using the Lyapunov. The ideal situation is to have the derivative of the Lyapunov function be negative definite so that the system can have guaranteed stability and convergence. Once the controller is designed with the estimated parameters, it is then plugged into the Lyapunov function and the adaptation law is found.

When developing the adaptive controller for a given system, the parameters in the controller are estimations of the actual parameters. Since parameters may change over time or may be completely unknown to begin with, the controller will “adapt” to them. The adaptation law uses the output of the system to send updated estimations of the parameters into the controller while the system is running.
Having a controller that is robust enough as the SMCPE and has the on-line adaptation capabilities due to adaptive control, the robust adaptive controller should be able to successfully adapt to unknown or changing parameters while rejecting external disturbances.

Figure 10 - Adaptive Control [21]
3 Data & Analysis
The vibration control project was broken down into five main sections. The first step was to derive the mechanical model of the system and the initial analysis of the passive suspension systems. Secondly, linear controllers such as PID and LQR controllers were developed and implemented. Since a more complex model requires an advanced controller, robust adaptive controllers were then designed. Finally, all of these controllers were tested on models of varying complexity.

3.1 Simulation

3.1.1 Model Derivation
Before development of the control law, the derivation of the mechanical model of the system had to take place. As was discussed in the previous section, using distributed-parameters is a more accurate form of modeling, but a discrete system is much easier to develop a controller around. To compromise, a continuous system was modeled in ANSYS and was compared to the discrete system in MATLAB.

The corresponding discrete system in MATLAB was derived from a distributed-parameters system, where the equivalent terms of a lumped-parameters system were found using the energy method. To find the equivalent mass of a clamped-free-clamped-free thin plate, the kinetic energy was determined.

\[
T = \frac{1}{2} \int_V \rho \left( \frac{\partial w(x,y,t)}{\partial t} \right)^2 dV
\]  

(33)
Then it was compared to the kinetic energy of a corresponding lumped-parameters system.

\[ T_e = \frac{1}{2} m_e \left[ \dot{\upsilon} \left( \frac{a}{\sqrt{2}} \right)^2 \right] \]  

(34)

Comparing these two equations gives:

\[ m_e = \rho h \int_0^x \int_0^y \frac{\varphi^2(x,y)dx dy}{\varphi^2(a/\sqrt{2},b/\sqrt{2})} \]  

(35)

Where \( \varphi(x,y) \) represents the eigenfunction of the plate at the location \( (x,y) \) on the plate. In this case, the point of interest is at the center of the plate, so the location is \( (a/\sqrt{2},b/\sqrt{2}) \).

This correlation is repeated for the equivalent stiffness term using the potential energy of the plate. Using the stress-strain relationships, the potential energy is given as:

\[ U = \int \left( \sigma_{xx}\varepsilon_{xx} + \sigma_{yy}\varepsilon_{yy} + \tau_{xy}\varepsilon_{xy} \right) dV \]  

(36)

The corresponding potential energy of a lumped-parameters system is:
When these two equations are compared as was the case for the equivalent mass, the resulting equivalent stiffness is:

\[ k_e = D \left[ \frac{\int_0^x \int_0^y \nabla^4 \varphi(x,y) \, dx \, dy}{\varphi^2 \left( \frac{a}{2}, \frac{b}{2} \right)} \right] \]  

where

\[ D = \frac{E h^3}{2(1 - \nu^2)} \text{ and } \nabla^4 = \frac{\partial^4 \varphi}{\partial x^4} + 2 \frac{\partial^4 \varphi}{\partial x^2 \partial y^2} + \frac{\partial^4 \varphi}{\partial y^4} \]  

To determine the value of the equivalent damping parameter a transient analysis was run, and the reduction in work over the course of one period corresponded to the equivalent damping term.

\[ W_{visc} = \int_0^\tau f_c \, dx, \quad \text{where } f_c = c \omega(x,y,t) \]  

The equivalent work due to viscous damping in a lumped-parameters system can be represented as:

\[ W_{visc} = \pi c_{eq} \omega A^2 \]  

where \( A \) is the amplitude at the point of interest and \( \omega \) is the frequency of oscillations. Therefore, the equivalent damping parameter comes to be:

\[ c_{eq} = \frac{W_{visc}}{\pi \omega A^2} \]  

Using these three equivalent parameters, the model can be rewritten as a discrete system as seen in the image below.
In this equivalent model, the damping term is small as it only takes into account viscous damping. Due to this, a gain was applied to the damping term such that it represented other losses of energy in the system such as frictions and other types of damping. When the transient response of these two models to a sinusoidal disturbance was compared, they matched in signature – that is they matched in frequency and phase. The detailed calculations of the eigenfunctions were done in both MATLAB and Maple, and are included in Appendix A – Equivalent Parameters Calculations (MATLAB and Maple). However, as the ANSYS simulation was looked at closer, it was seen that there exists nonlinear effects. These nonlinearities in the model could be due to internal damping, or even boundary conditions of the system.
Also, the geometry has a significant impact on the analysis of a continuous system. As seen below, the difference in location on a plate from the corner to the middle has noticeably different response amplitude.

![ANSYS Geometry](image)

**Figure 14 - ANSYS Geometry**

From this point, it was determined that the system model could be well represented by the ANSYS model. Although the nonlinearities do not show in the MATLAB environment, the modal analysis from ANSYS could be used to closely represent the model of the system. In turn, the advanced controller needed to account for these nonlinearities and unmodeled dynamics.

### 3.1.1.1 ANSYS – MATLAB Interface

Once the analysis of the plate was matched between discrete and continuous systems, the geometry was expanded to represent the entire system in the form of a box. The updated ANSYS model with the new geometry is shown below in Figure 15.
Although the calculated equivalent model proved it matched the signature of ANSYS simulations, the modal analysis that was done in ANSYS better represented the system as a lumped-parameters model taking into account all energy loss such as damping, boundary conditions, unmodeled dynamics, etc. It does this through the damped frequency determined by modal analysis in ANSYS as will be discussed below.

Running a modal analysis of the box allowed for the development of a discrete system to be used in MATLAB and Simulink. The modal analysis of a plate representing the base of the box was first done. Using D.J. Gorman’s *Free Vibration Analysis of Rectangular Plates*, the first 5 natural modes of the plate could be determined [23]. The values of lambda for the first five modes of the plate depended on the geometry and material of the plate. Once the ratio was determined and the lambda values were found, the natural frequencies were found by the equation:

\[ \lambda^2 = \omega a^2 \sqrt{\frac{\rho}{D'}} (\nu = 0.333) \]  

(43)

When the ANSYS simulation was run for the same dimensions and material properties, there was still a difference between the two frequencies. This was taken to be the unmodeled dynamics of the system. The difference could be
accounted for by bending and torsional modes as well as structural damping within the material. Once the unmodeled dynamics of the system were determined, the damping coefficient was held constant for the same material and mode while expanding to a more complex geometry. For example, the damping ratio of a plate the size of the base of the aluminum box was held constant while the geometry expanded from the base of the box to the entire box. This allowed for the damped frequency to be determined through the ANSYS simulation of the complex geometry, and from that the natural frequency was found using the damping ratio. After this was determined for the first mode, the system was assumed to be classically damped so the relation \( \zeta \omega_n \) remained constant for latter modes of the object.

Initially, the base of the aluminum box was compared between ANSYS and hand calculations. Using equation (11) above and the properties of aluminum, the natural frequencies of the plate were determined to be 3810.7Hz, 33972.4Hz, 69368.8Hz, etc. Focusing on the first fundamental frequency, the corresponding ANSYS calculation was 2105.5Hz. This is where the damping ratio, \( \zeta \), was determined by the equation:

\[
\omega_d = \omega_n \sqrt{1 - \zeta^2}
\]

Once the damping ratio was determined for a simple plate, the ANSYS simulation was expanded to do the modal analysis of the entire aluminum box. This resulted in a damped frequency of 3168.9Hz for the first mode. Using the equation above, the natural frequency of the aluminum box was found (\( \omega_{n1} = 4407.3Hz, \zeta_1 = .6947 \)) and the simplified equation of motion of the aluminum box could be represented by,

\[
\ddot{x} + 2\zeta \omega_n \dot{x} + \omega_n^2 x = u + d
\]

When the controllers were designed for a one-mass system, this is the equation of motion that was used.
3.1.2 Passive Suspension Models
Before an active controller was developed, the comparison of passive suspension systems had to be explored. The three passive systems that were simulated and analyzed the most are shown below.

![Passive Suspension Systems](image)

**Figure 16 - Passive Suspension Systems**

To test the three passive suspensions shown above, they were simply compared as a thin plate supported at four points by a combination of spring, damper, and friction elements as shown in Figure 17. In Simulink this was represented by a lumped mass determined by volume and density of the plate, and the passive components corresponded with the spring, damping, and friction elements implemented in ANSYS.

![Continuous Mass, Spring, Damper System](image)

**Figure 17 - Continuous Mass, Spring, Damper System**
The discrepancy above was been due to the nonlinear effects within the flexible structure, as well as the location on the structure as described in the previous section 3.1.1. The second suspension model that was designed and simulated was the spring in parallel with the Maxwell arm as shown in the middle of Figure 16. The Maxwell arm consists of the spring and damper in series. This specific orientation resulted in similar discrepancies between the ANSYS and MATLAB models as identified in the first suspension. The steady state response of the second suspension design has the ANSYS analysis with slightly higher amplitude than that from MATLAB. This is shown in Figure 19 below.
The third passive suspension model tested was a combination of a spring, damper and dry friction in parallel. This dry friction can also be referred to as coulomb damping, and represents nonlinearity in the suspension. Again the Simulink and Simscape show the same result while the ANSYS is different.
After these three passive suspension models were compared in depth, other variations using these same components were simulated. The other passive models and the corresponding results are shown below.

![Figure 21 - Suspension Models](image1)

![Figure 22 - Suspension Variation Comparison](image2)
As seen from the last plot in Figure 22 and the corresponding models in Figure 21, the only suspension model that could not be simulated was number 8. Out of the rest of the models compared, only suspension model number 9 diverged. It could be noticed that both of these models had coulomb damping components in series with either spring or damper components.

All of the different models were compared while varying the stiffness and damping parameters. This showed the effects that the passive components of a controller could provide for the system in case there was some type of active controller failure.

### 3.1.3 Active Linear Controller Design
Since there will be unknown disturbances acting on the system and the materials and parameters of the system may vary, a passive suspension system is not adequate. In this case active linear controllers were designed and compared. The first linear controller implemented in the system was a PID controller. In the design of this controller, the proportional, integral, and derivative gains were adjusted to find the best performance of the system. Of course, performance comes at a cost. The effort from the controller is also monitored and taken into account as discussed in the Vibration Control section.

Before any suspension models can be analyzed or any controllers were designed, the dynamics of the system were derived. In the case of a simple mass, spring, damper system, the equation of motion came to be:

\[
m\ddot{x}(t) + c\dot{x}(t) + kx(t) = u(t) + d(t) \tag{46}
\]

Where \( u(t) \) is the controller input to the system, and \( d(t) \) is the external disturbance applied on the system. The disturbance was set to zero and a desired reference was assigned to 0.01 for gain optimization. This equation of motion can also be rewritten in the form seen above in equation 45.

#### 3.1.3.1 PID Controller Design
This PID controller was tested and designed for a simple rectangular plate which is the equivalent to a mass spring, damper system, where the parameters were
determined from an ANSYS simulation. The Simulink diagram of the system with PID is shown below in Figure 23

![Simulink Diagram](image)

**Figure 23 - System with PID Controller**

As seen below in Figure 24, adjusting the proportional, integral, and derivative gains lead to an improved performance (negative direction) for an equivalent increase in controller effort. However, it can be noticed that for the derivative gain, the controller effort increases much quicker than for the proportional or integral gains. Also, increasing the integral gain increases the effort as well as improves the performance slower than the proportional gain. There is a tradeoff here between the two, and it is more beneficial to adjust the proportional gain first, then the integral gain, and lastly the derivative gain.
Using the tuning feature in MATLAB, the PID constants were adjusted to find the best combination. The table below shows the gains the tuner chose and the corresponding performance and effort of the controller.

**Table 1 - PID Gain Tuning**

<table>
<thead>
<tr>
<th>P</th>
<th>I</th>
<th>D</th>
<th>CE</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.40E-03</td>
<td>1.00E-04</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>1</td>
<td>9.37E-02</td>
<td>1.00E-04</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>1</td>
<td>3.49E-01</td>
<td>1.00E-04</td>
</tr>
<tr>
<td>1</td>
<td>500</td>
<td>1</td>
<td>8.39E+00</td>
<td>1.00E-04</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>1</td>
<td>3.34E+01</td>
<td>1.00E-04</td>
</tr>
<tr>
<td>1</td>
<td>10000</td>
<td>1</td>
<td>3.33E+03</td>
<td>1.00E-04</td>
</tr>
<tr>
<td>1</td>
<td>100000</td>
<td>1</td>
<td>3.33E+05</td>
<td>9.99E-05</td>
</tr>
<tr>
<td>1</td>
<td>7.50E+05</td>
<td>1</td>
<td>1.87E+07</td>
<td>9.96E-05</td>
</tr>
<tr>
<td>1.00E+03</td>
<td>1.00E+06</td>
<td>1</td>
<td>3.33E+07</td>
<td>9.94E-05</td>
</tr>
<tr>
<td>1</td>
<td>5.00E+06</td>
<td>1</td>
<td>8.15E+08</td>
<td>9.71E-05</td>
</tr>
<tr>
<td>1</td>
<td>5.00E+10</td>
<td>1</td>
<td>2.78E+12</td>
<td>1.71E-07</td>
</tr>
<tr>
<td>1.00E+08</td>
<td>6.00E+12</td>
<td>1</td>
<td>Diverges</td>
<td></td>
</tr>
</tbody>
</table>
From the table above, it is seen that the control effort increases exponentially for even the slightest improvement in performance of the system.

Using the gain values \( P=2e7, I=1e8, D=1e6 \), the system was tested for different disturbances. For a disturbance of \( A\sin \omega t \), where \( A=0.010m \), and \( \omega=100Hz \), the system with a PID controller is compared with the same system with simple feedback shown below.

![Figure 25 - System Response, PID (blue) vs Simple Feedback (red)](image)

The PID controller minimized the effect the disturbance had on the system, but it did not allow for complete rejection of the external disturbance.

### 3.1.3.2 LQR Design

Another linear controller that was explored for this system was the linear quadratic regulator (LQR) controller. This LQR was implemented around a state-space system, where the controller was a gain matrix \( K \).

\[
\dot{x} = Ax + Bu \quad y = Cx, \quad u = -Kx
\]  

(47)
Assuming that all the states can be measured and controlled, MATLAB can be used to calculate the optimal gain matrix. The figure below shows the system with the LQR controller implemented.

By modifying the proportional constant $\rho$, the importance of the controller is developed around minimizing the controller effort or performance. The following plot shows that as the value of $\rho$ is changed, both the effort and performance of the controller vary.

Figure 26 - System with LQR Controller
The LQR controller is efficient for systems that have known parameters and no external disturbances. As seen below, the LQR controller can achieve set point tracking.
In Figure 29 it is shown that the LQR minimizes the vibrations of the system but the external disturbance is still present in the resulting position of the mass. In this case, the LQR is not robust enough to completely reject the disturbances.

![Figure 29 – LQR Set Point Tracking, w/Disturbance](image)

This optimal controller was designed assuming that all the states of the system can be measured and controlled. However, it is not always the case that all states are known. If the states are unknown, an observer can be designed for the system. This observer can estimate the states and use the estimates in the feedback loop instead of the real states which are immeasurable. The equation of this observer is:

$$\dot{x} = A\hat{x} + Bu + L(y - \hat{y})$$  \hspace{1cm} (48)

Where $y - \hat{y}$ is the output estimation error. The Simulink diagram that includes the observer is shown in Figure 30.
When the simulation is run with the observer driving the feedback loop, it is seen that the estimated results exactly match the real results. Figure 31 below corresponds directly with Figure 28 showing the position tracking using the LQR method.

3.1.4 Robust Adaptive Controller
As seen by the results given, the PID and LQR controllers minimize the effect of the external disturbances to the system. However, when using these controllers it
is assumed that all the parameters of the system are known. This is not always the case, as parameters can vary over time. One example of a time varying parameter is that repetitive strain in a material can change the equivalent spring constant. Also, there could be frictions or nonlinearities in the system due to the way something is clamped or attached. Unfortunately, these unmodeled dynamics cannot be modeled in the programs. Even if there are nonlinear components added into the plant of the system, the program can read the plant and the PID or linear controller can adapt to the dynamics. Also, when the parameters are altered in the simulations, it is known to the program, hence the ability for the PID to work without a problem. In some cases it has an even better response regarding controller effort and performance than that of the robust adaptive controller. In essence, if there are no unmodeled dynamics in the experiment, the PID controller should work. However, when the experiment is set up and data is taken, it will be seen that this is not the case. The linear controllers don’t account for these changes, which is why it is necessary to design an advanced controller that can adapt to these unmodeled dynamics and be robust enough to reject external disturbances.

Using the same equation of motion derived above for a simple mass, spring, damper system a sliding mode controller was designed. The first step in design of a sliding mode controller was to choose the sliding hyperplane.

$$s(t) = \dot{e}(t) + \lambda e(t), \quad e(t) = x_d(t) - x(t)$$  \hspace{1cm} (49)

The error is defined as the difference between the desired trajectory, $x_d(t)$ and the actual position of the object $x(t)$. The gain $\lambda$ is a weighting factor for the error of the system. Next, the controller is designed that will guarantee stability and convergence of the system. Taking the derivative of the sliding hyperplane, $s(t)$, and substituting the equation of motion results in the equation below.

$$\dot{s}(t) = \ddot{x}_d - \frac{1}{m} (u + d - c \dot{x} - kx) + \lambda (\dot{x}_d - \dot{x})$$  \hspace{1cm} (50)

From here, the controller can be defined.
\[ u(t) = \hat{m}(\dot{x}_d + \lambda(\dot{x}_d - \dot{x}) + a_1 s + a_2 s \text{sgn}(s)) + \ddot{c} \dot{x} + \dddot{k} x - \ddot{d}(t) \] (51)

The controller takes into account the derivative of the sliding hyperplane and adds some terms to make the derivative of the Lyapunov function negative definite.

The next step is to choose a Lyapunov function that has energy-like terms.

\[ V(t) = \frac{1}{2} m s^2 + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}, \quad \Gamma = \begin{bmatrix} y_1 & 0 & 0 \\ 0 & y_2 & 0 \\ 0 & 0 & y_3 \end{bmatrix}, \tilde{\theta} = \begin{bmatrix} \hat{\epsilon} \\ \ddot{c} \\ \ddot{k} \end{bmatrix} \] (52)

The derivative is then taken such that:

\[ \dot{V}(t) = m s \dot{s} + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \] (53)

After many calculations and manipulations, the derivative of the Lyapunov simplifies to:

\[ \dot{V}(t) = \hat{m}(-a_1 s^2 - a_2 s \ast \text{sgn}(s)) + \tilde{\theta}^T \left( Y^T s + \Gamma^{-1} \ddot{\theta} \right) - s(d - \dot{d}) \] (54)

where,

\[ Y = \{ \dot{x}_d + \lambda(\dot{x}_d - \dot{x}) \}_{1 \times 2} \]

\[ \ddot{d}(t) \triangleq \psi_{\text{est}}(t) = \hat{m}\ddot{x} + \dddot{c} \dot{x} + \dddot{k} x - u(t - \tau), \quad \ddot{x}(t) = \frac{\dot{x}(t) - \dot{x}(t - \tau)}{\tau} \] (56)

Since it is assumed that the perturbation estimation, \( \psi_{\text{est}} \), is so close to the actual disturbance, \( d(t) \), for a significantly small time step, the last term in the derivative of the Lyapunov is said to go to zero. Thus it leaves,

\[ \dot{V}(t) = \hat{m}(-a_1 s^2 - a_2 s \ast \text{sgn}(s)) + \tilde{\theta}^T \left( Y^T s + \Gamma^{-1} \ddot{\theta} \right) \] (57)

The coefficient \( \hat{m} \) of is always negative definite, so the coefficient of \( \tilde{\theta}^T \) has to go to zero to guarantee stability and convergence. This allows the adaptation laws to be found after some further calculation and manipulation:

\[ \hat{m} = \hat{m}_0 + \int_0^t y_1(\dot{x}_d(\tau) + \lambda(\dot{x}_d(\tau) - \dot{x}(\tau)))(\dot{e} + \lambda e) d\tau \] (58a)

\[ \dot{\hat{c}} = \dot{\hat{c}}_0 + \int_0^t y_2 \dot{x}(\tau)(\dot{e} + \lambda e) d\tau \] (58b)
\[ \dot{k} = \dot{k}_0 + \int_0^t \gamma_3 x(\tau)(\dot{e} + \lambda e) d\tau \]  

(58c)

Once the sliding hyperplane, control law, adaptation law, and perturbation estimation are all found and determined, they have to be implemented surrounding a plant for simulation. This Simulink diagram is shown in Figure 30 below.

![Figure 30 - System with Robust Adaptive Controller](image)

After the controller was implemented, there were many steps taken to find the best design. The controller gains, \(a_1\), \(a_2\), and \(\lambda\) were adjusted to find the best performance of the system while monitoring the amount of effort it was taking the controller to stabilize the system.
In addition to the controller gains, the weighting factors, $\gamma$, for the adaptation law were tested. However, these values did not have any impact on either the performance of the system or the controller effort.

Another design consideration was determining which switching function to use in the sliding mode controller. The controller was initially tested using the signum switching function. However, as seen below in Figure 34 the signum function can cause chattering in the system. This chattering can often excite higher modes of the system making it unstable.
Another switching function that was considered was the tangent hyperbolic function. This function was implemented to smoothen the system response and eliminate chattering. The transient response using the tangent hyperbolic switching function is shown below in Figure 35.
Once it was determined that the tangent hyperbolic switching function successfully eliminated the chattering, the constants $\alpha$, $\beta$, and $\lambda$ were again tested. This time, the response of the function to varying gains had a more smooth and predictable output. The gains $a_1$ and $a_2$ have the exact same response to increasing value – an improvement in performance for a larger effort from the controller. The gain $\lambda$ also shows an improvement in performance at the expense of controller effort, but on a larger scale than $a_1$ and $a_2$. The response of the system to varying gains can be shown in Figure 36.

![Robust Adaptive Controller Gains (Tangent Hyperbolic)](image)

**Figure 36 - RA Controller Gains (tanh)**

The gains were set to the optimal value based on the gain tests. After the gains and switching function were determined, the controller was shown to be both robust and adaptive. To prove adaptability, the values of the plant parameters were altered, and the program was run to see the controller adjust to the unknown values.
Table 2 - Adaptability of Robust Adaptive Controller

<table>
<thead>
<tr>
<th>% Change</th>
<th>CE</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00E+00</td>
<td>4.67E+13</td>
<td>5.99E-05</td>
</tr>
<tr>
<td>5.00E+00</td>
<td>5.12E+13</td>
<td>6.28E-05</td>
</tr>
<tr>
<td>1.00E+01</td>
<td>5.59E+13</td>
<td>6.58E-05</td>
</tr>
<tr>
<td>1.50E+01</td>
<td>6.08E+13</td>
<td>6.88E-05</td>
</tr>
<tr>
<td>2.00E+01</td>
<td>6.58E+13</td>
<td>7.18E-05</td>
</tr>
<tr>
<td>2.50E+01</td>
<td>7.09E+13</td>
<td>7.48E-05</td>
</tr>
<tr>
<td>3.00E+01</td>
<td>7.63E+13</td>
<td>7.77E-05</td>
</tr>
<tr>
<td>3.50E+01</td>
<td>8.18E+13</td>
<td>8.07E-05</td>
</tr>
<tr>
<td>4.00E+01</td>
<td>8.74E+13</td>
<td>8.37E-05</td>
</tr>
<tr>
<td>4.50E+01</td>
<td>9.32E+13</td>
<td>8.67E-05</td>
</tr>
<tr>
<td>5.00E+01</td>
<td>9.92E+13</td>
<td>8.97E-05</td>
</tr>
</tbody>
</table>

In the chart above, the parameters of the model were varied from zero to 50 percent and the corresponding effort and performance of the controller were recorded. As expected, the farther the parameters get from the estimated values in the adaptation law, the more effort the controller uses to achieve tracking. It also results in a worse performance because it takes longer to converge to the desired trajectory.

To test robustness, the external disturbance applied to the system was altered. The simple narrowband input \( d(t) = A \sin \omega t \) was applied as well as broadband inputs and white noise. The results of these simulations are shown in Table 3 below.

Table 3 - Robustness of Robust Adaptive Controller

<table>
<thead>
<tr>
<th>Disturbance Type</th>
<th>CE</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>5.27E+11</td>
<td>7.71E-04</td>
</tr>
<tr>
<td>Broadband</td>
<td>5.56E+11</td>
<td>7.67E-04</td>
</tr>
<tr>
<td></td>
<td>552.10%</td>
<td>56.16%</td>
</tr>
<tr>
<td>Narrowband</td>
<td>5.53E+11</td>
<td>7.64E-04</td>
</tr>
<tr>
<td></td>
<td>495.71%</td>
<td>91.57%</td>
</tr>
<tr>
<td>Narrowband+Broadband+</td>
<td>5.73E+11</td>
<td>7.60E-04</td>
</tr>
<tr>
<td>White Noise</td>
<td>875.23%</td>
<td>136.57%</td>
</tr>
</tbody>
</table>

As seen above, when the disturbances were added to the system, the controller needed to put in much more effort to improve the performance of the system.
relative to the simulations with no disturbances. Even with all the disturbances applied and the parameters of the plant varied 50% from the estimated values in the adaptation law, the controller is able to learn the system and track the desired trajectory. This is shown below in Figure 37.

![RA Tracking Problem with \(d(t)\) and Unknown Parameters](image)

Figure 37 - RA Response to Disturbance & Varying Parameters

Just as the linear controllers can adapt to the dynamics of the system during simulations, the robust adaptive controller developed for a one-mass system can adapt to a varying system structure during the simulations. When the plant is completely unknown, the controllers will not have as good an estimation of the system and therefore will not work as well. This is where the advanced robust adaptive controller comes into play. The controller will allow for adaptation to a complex plant while also being able to control a simpler plant with less effort. Essentially, it is easier to have an advanced system control a simpler plant, than vice versa.

### 3.1.4.1 Advanced System Model

Once the robust adaptive controller was developed for the single mass system, the model was expanded to better represent the true system. The system equations of
motion were re-derived and an advanced controller was developed for a system consisting of two masses. The goal of this model was to eliminate the vibrations to the second mass while only having the ability to control the first mass. To do this, the equations of motion were manipulated and the first state of the system was substituted for so that the second state was in terms of the control law. For this model, the system had to be nondimensionalized in order to simulate it. The first step was to derive the equations of motion.

After nondimensionalizing using the relation \( \tau = Gt \), these equations of motion become:

\[
m_1 \ddot{x}_1(t) + c_1 \dot{x}_1(t) + c_2 (\dot{x}_1(t) - \dot{x}_2(t)) + k_1 x_1(t) + k_2 (x_1(t) - x_2(t)) = u(t) + d(t)
\]

\[= (59a)\]

\[
m_2 \ddot{x}_1(t) - c_2 (\dot{x}_1(t) - \dot{x}_2(t)) - k_2 (x_1(t) - x_2(t)) = 0
\]

\[= (59b)\]

After nondimensionalizing using the relation \( \tau = Gt \), these equations of motion become:

\[
m_1 \ddot{x}_1 + G c_1 \dot{x}_1 + G c_2 (\dot{x}_1 - \dot{x}_2) + G^2 k_1 x_1 + G^2 k_2 (x_1 - x_2) = G^2 u + G^2 d
\]

\[= (60a)\]

\[
m_2 \ddot{x}_1 - G c_2 (\dot{x}_1 - \dot{x}_2) - G^2 k_2 (x_1 - x_2) = 0
\]

\[= (60b)\]

where \( x_1(t) = x_1(\tau) \), etc.

In order to remotely control the second mass of the system, the equations of motion were simplified by eliminating the damping terms. After substitution and simplification, the modified equation of motion was,

\[
M x_2^{(4)} + C \dot{x}_2 + K x_2 = G^2 u + G^2 d
\]

\[= (61)\]
where, \( M = \frac{m_1 m_2}{G^2 k_2} \), \( C = \left( m_1 + \frac{m_2 k_1}{k_2} + m_2 \right) \), and \( K = (G^2 k_1) \)

Once the equation of motion was derived such that the second mass was related to the controller, the sliding hyperplane was determined to be,

\[
s(\tau) = \ddot{x} + \lambda_1 \dot{x} + \lambda_2 \dot{x} + \lambda_3 e, \quad e(\tau) = x_{d2} - x_2, \quad \dot{e}(\tau) = \dot{x}_{d2} - \dot{x}_2, \quad \text{... (62)}
\]

Just as in the robust adaptive controller for the single mass system, the derivative is taken of the sliding hyperplane and the control law is derived.

\[
u(\tau) = \frac{1}{G^2} \left( \dot{\lambda} \ddot{x}_2 + \dot{\lambda} x_2 + \mu \left[ x_{d2} + \lambda_1 (\ddot{x}_{d2} - \ddot{x}_2) \right] \right) \left[ \lambda_2 (\dot{x}_{d2} - \dot{x}_2) + \lambda_3 \dot{x}_2 \right] + \nu + \beta \text{sgn}(\nu) \quad \text{... (63)}
\]

In this case, the perturbation estimation is determined from the modified equation of motion to be:

\[
\dot{\nu}(\tau) = \psi_{est}(\tau) = \frac{1}{G^2} \left[ \dot{\mu} x_2 + \dot{\mu} \ddot{x}_2 + \dot{\mu} x_{d2} - \mu \left( \tau_\nu - \tau_{\nu-1} \right) \right] \quad \text{... (64)}
\]

where \( \tau_{\nu-1} \) represents the previous timestep.

Using the same Lyapunov equation as above and setting the coefficient of \( \dot{\mu} \) to zero to prove stability results in the adaptation law.

\[
\dot{\mu} = \dot{\mu}_0 + \int_0^\tau \gamma_1 \left[ x_{d2} + \lambda_1 (\ddot{x}_{d2} - \ddot{x}_2) \right] s \, d\tau \quad \text{... (65a)}
\]

\[
\dot{\lambda} = \dot{\lambda}_0 + \int_0^\tau \gamma_2 \ddot{x}_2 s \, d\tau \quad \text{... (65b)}
\]

\[
\dot{R} = \dot{R}_0 + \int_0^\tau \gamma_3 x_2 s \, d\tau \quad \text{... (65c)}
\]

This controller is in the process of being developed. Just as in the one-mass system, the controller gains will be tested, as well as the adaptability and robustness of the system.
The benefit of these robust adaptive controllers is that they can account for unmodeled dynamics or unknown parameters within the system. They can also reject external disturbances by using perturbation estimation and adapting to the environment. This is something that the linear controllers are not capable of handling. It is important not just to have a controller capable of handling all the unknowns of the system, but to have an accurate representation of the model to develop the controller. For example, the first robust adaptive controller that was designed was developed around a 1 mass system. Since it is a robust adaptive controller, it can adjust to a plant that may have two masses. However, in doing so it will require extensive effort from the controller to learn the new system and minimize the disturbances. By having a more detailed model, the second robust adaptive controller was designed around this two mass system. In this case, if the system is any simpler than what the controller was designed for, it will take less effort to adjust to the new plant.

3.2 Experimentation
Since all the theory is done, the next step is to implement the controllers on real systems. The experimental portion of this project will take many additional steps including design of the parts and drawings, fabrication and assembly before testing is even started. Testing will consist of optimizing controller parameters, and gathering data to support the simulations and theory discussed above.

3.2.1 Part Design and Assembly
The experiments will repeat the steps taken in the simulations. The first step will be to actuate a thin plate to induce vibrations. To do this, an adapter block was made that will distribute the disturbance to four inputs in the corners of the system. The inertial actuator will excite the adapter block which will then excite the plate of interest representing the system. A diagram of this setup is shown below.
The four spacers between the adapter block and the plate act as the passive suspension between the input of the disturbance and the system. The inertial actuator will be the “unknown” source of vibrations that has to be controlled. After the plate is excited by the first actuator, a second inertial actuator acting as an absorber will be mounted to the plate. Now the setup will consist of two actuators – one as a disturbance, another as an active vibration absorber. Once the system is running in this configuration, the plate can be substituted out for a real representation of the system in the form of a box. The proposed experimental setup is shown in Figure 40 below.
3.2.2 Experimental Setup
The experimental setup will consist of a disturbance and a control voltage going into the system, with a Laser Vibrometer sensor providing feedback in the form of the displacement of the plate and box. The disturbance will be produced by an oscilloscope which will send an “unknown” signal to the model. The Laser Vibrometer from Polytec will be mounted above the model and therefore be able to capture the displacement of the box. This measurement will be fed back into the dSPACE system, where it will be compared against the desired displacement – in this case the displacement is zero because it is a regulation problem. The controller will then use this information along with estimated parameters and disturbances to produce a signal that will suppress the vibrations of the model. The dSPACE will send a signal through the amplifier to the system, and again the displacement will be read by the Laser Vibrometer thus completing the feedback loop. Below is a flow chart that describes the entire system.
Figure 41 - Experimental Setup
4 Conclusion

4.1 Summary and Discussion

The vibrations of flexible structures are seen in many industrial applications, whether it is robotics, space systems, or another field. The robust adaptive controller modeled in this thesis guarantees stability to systems like these despite changing conditions in the systems or external disturbances. In order to derive this control law, the model was successfully reduced from a continuous system to an equivalent discrete system to be used for controller design.

Using the eigenfunction of a fully clamped plate and the energy method, the corresponding distributed-parameters system could be represented by a lumped-parameters mass, spring, damper system. The equivalent calculations used the eigenfunction of the plate in terms of the overall dimensions as well as the eigenfunction evaluated at a specific location on the plate. For the purposes of this model, the location of interest was taken to be the center of the plate. The equivalent model matched the ANSYS simulation for a distributed-parameters system in signature and phase, but the nonlinear term that represented the loss of energy in the system had a gain applied in the lumped-parameters model in order to match amplitudes. The gain applied to the damping term represented the loss of energy in the system due to unmodeled dynamics. The unmodeled dynamics in this case could comprise of different boundary conditions, extra damping, or other nonlinear effects. In this case, the system was modeled as a box and held at the four corners. So in addition to different boundary conditions, the system geometry didn’t match that of a simple plate.

Knowing there was a well derived model of the system, a controller was designed to stabilize it. When finding the best controller to eliminate vibrations to the system, it is crucial to take into account all aspects of the system. Linear controllers such as PID and LQR are good for systems in which the parameters are known values and don’t change over time. However, systems that involve multiple components often have nonlinear terms associated with mounting and
clamping. Also, most systems rarely have parameters that hold constant through the entire period of use. It is common to have materials that deteriorate over the life of a product in addition to connections within a system that can loosen during repeated exposure to loads and vibrations. With all these unknowns in the system it is necessary to account for unmodeled dynamics in addition to being about to reject external disturbances. The effect of these unmodeled dynamics and varying parameters cannot be seen in simulations, because ultimately, the program can still read into the “hidden” plant. This is where the need for a more advanced controller lies.

After many passive and active linear controller designs were explored, it was deemed necessary to create a more advanced controller. This robust adaptive controller was able to reject external disturbances and guarantee stability in the presence of unmodeled dynamics. The controller was also able to account for varying parameters.

During the experiment, it will be seen that the linear controllers cannot control a complex system. Although they have very good response in the simulations even to changes in the system, that does not correlate to the unknown of the real experimental model. The robust adaptive controllers on the other hand, will have the ability to adapt to varying parameter values within the system while accounting for unmodeled dynamics such as linearities and rejecting the external disturbances on the system.

The accuracy of the derivation of the system model plays an important role when designing the controller. A more detailed model with multiple masses will complicate the development of the controller, but allow for better performance at much less effort. When the controller made for a single mass system tried to control a more complex plant, it was able to adapt to the new model and track a desired trajectory. However, this came at a very high cost, as the controller had to work very hard to achieve good performance. Contrary to this, the advanced robust adaptive controller designed around a complex plant will be able to control
the single mass plant with less increase in effort from the original model it was
designed for.

Different gains were explored for all controllers, and in the robust adaptive
controller the switching function played a significant role. It was seen that the
signum function created a chattering effect, while the tangent hyperbolic function
was implemented to smoothen this response. In all controllers, the gains directly
affected the performance of the system at the expense of the effort of the
controller.

4.2 Future Work

The next step is to implement the controllers on a physical experiment. Currently,
the setup allows for a plate representing the system to be excited. Once this is
shown, the model will be modified to include an inertial actuator that acts as the
vibration absorber as well as a better representation of the system. The different
controllers will then be implemented on both the plate and box.

It will be shown that with a good model, the effort of the controller decreases. It
will also be shown that linear controllers will not be able to handle the varying
parameters, unknown disturbances or unmodeled dynamics of the system. This is
where the robust adaptive controller with perturbation estimation will come into
play. During the implementation of each controller, the associated gains will be
optimized to achieve the best result. This analysis can then be compared to the
simulations and to other controllers.
5 Appendices

5.1 Appendix A – Equivalent Parameters Calculations (MATLAB and Maple)

The Maple code derives the eigenfunction evaluated at each boundary condition before the constants are determined.
From this point, the 8 boundary condition equations were brought into MATLAB in order to solve and simplify. The m-file is included below.

clear
clc

%L = .0635;
%Wb = .0508;
E = 68.9e9;
nu = 0.33;
rho = 2700;
t = .003175;
D = ((1/12)*(E*t^3)/(1-nu^2));

%Eigenvalues based on BC's for fully clamped plate (Gorman pg 93)
beta(1)=sqrt(7.472);
beta(2)=sqrt(22.31);
beta(3)=sqrt(31.88);
beta(4)=sqrt(45.32);
beta(5)=sqrt(50.52);

%Boundary condition equations from Maple
syms x y C1 C2 C3 C4 C5 C6 C7 C8
BC1 = C3*sin(b1*y)+C4*cos(b1*y)+C7*sinh(b2*y)+C8*cosh(b2*y);
BC2 = C1*a1*sin(b1*y)+C2*a1*cos(b1*y)+C5*a2*sinh(b2*y)+C6*a2*cosh(b2*y);
BC3 = C1*sin(a1*L)*sin(b1*y)+C2*sin(a1*L)*cos(b1*y)+C3*cos(a1*L)*sin(b1*y)+C4*cos(a1*L)*cos(b1*y)+C5*sinh(a2*L)*sinh(b2*y)+C6*sinh(a2*L)*cosh(b2*y)+C7*cosh(a2*L)*sinh(b2*y)+C8*cosh(a2*L)*3*cosh(b2*y);
BC4 = C1*cos(a1*L)*a1*sin(b1*y)+C2*cos(a1*L)*a1*cos(b1*y)-C3*sin(a1*L)*a1*sin(b1*y)-C4*sin(a1*L)*a1*cos(b1*y)+C5*cosh(a2*L)*a2*sinh(b2*y)+C6*cosh(a2*L)*a2*cosh(b2*y)+C7*sinh(a2*L)*a2*sinh(b2*y)+C8*cosh(a2*L)*2*cosh(b2*y)*sinh(a2*L)*a2;
BC5 = C2*sin(a1*x)+C4*cos(a1*x)+C6*sinh(a2*x)+C8*cosh(a2*x)^3;
BC6 = C1*sin(a1*x)*b1+C3*cos(a1*x)*b1+C5*sinh(a2*x)*b2+C7*cosh(a2*x)*b2;
BC7 = C1*sin(a1*x)*sin(b1*Wb)+C2*sin(a1*x)*cos(b1*Wb)+C3*cos(a1*x)*sin(b1*Wb)+C4*cos(a1*x)*cos(b1*Wb)+C5*sinh(a2*x)*sinh(b2*Wb)+C6*sinh(a2*x)*cosh(b2*Wb)+C7*cosh(a2*x)*sinh(b2*Wb)+C8*cosh(a2*x)*3*cosh(b2*Wb);
BC8 = C1*sin(a1*x)*cos(b1*Wb)*b1+C2*sin(a1*x)*sin(b1*Wb)*b1+C3*cos(a1*x)*cos(b1*Wb)*b1+C4*cos(a1*x)*sin(b1*Wb)*b1+C5*sinh(a2*x)*cosh(b2*Wb)*b2+C6*sinh(a2*x)*sinh(b2*Wb)*b2+C7*cosh(a2*x)*cosh(b2*Wb)*b2+C8*cosh(a2*x)*3*sinh(b2*Wb)*b2;

%Solving eigenfunction problem in terms of variables
for i=1:1,
    W1(i) = C1*sin(a1(i)*x)*sin(b1(i)*y)+C2*sin(a1(i)*x)*cos(b1(i)*y)+C3*cos(a1(i)*x)*sin(b1(i)*y)+C4*cos(a1(i)*x)*cos(b1(i)*y);
    W2(i) = C5*sinh(a2(i)*x)*sinh(b2(i)*y)+C6*sinh(a2(i)*x)*cosh(b2(i)*y)+C7*cosh(a2(i)*x)*sinh(b2(i)*y)+C8*cosh(a2(i)*x)*cosh(b2(i)*y);
end

%Total eigenfunction before finding C1-C8
W = W1+W2;

%Solving for constants
syms C1 C5

ggg=solve(BC1,BC2,BC3,BC5,BC6,BC7,'C2','C3','C4','C6','C7','C8')

C2 = ggg.C2;
C3 = ggg.C3;
C4 = ggg.C4;
%C5 = ggg.C5;
C6 = ggg.C6;
C7 = ggg.C7;
C8 = ggg.C8;

C1 = 1;
C5 = 1;

C2 = eval(C2);
C3 = eval(C3);
C4 = eval(C4);
C6 = eval(C6);
C7 = eval(C7);
C8 = eval(C8);

% Final eigenfunction
WW1 = eval(W1);
WW2 = eval(W2);
WW = WW1 + WW2;

L = 0.0635;
Wb = 0.0508;

phi = 0.0635 / 0.0508; % phi = b/a

for i = 1:2,
    a1 = 1/phi*sqrt(beta(1)^2*phi^2 + (pi/L)^2);
    b1 = 1/phi*sqrt(-beta(1)^2*phi^2 + (pi/L)^2);
    a2 = 1/phi*sqrt(beta(1)^2*phi^2 + (pi/Wb)^2);
    b2 = 1/phi*sqrt(-beta(1)^2*phi^2 + (pi/Wb)^2);
end

% Eigenvalue for specific location (x, y) on plate
%x = 0.03175; y = 0.0254;
%WW = eval(WW);

% Variations of eigenfunction for equivalent parameters
syms x y
WWdx = diff(WW, x);
WWddx = diff(WWdx, x);
WWdddx = diff(WWddx, x);
WWddddx = diff(WWdddx, x);
WWdy = diff(WW, y);
WWddy = diff(WWdy, y);
WWdddy = diff(WWddy, y);
WWddddy = diff(WWdddy, y);
WWdxdy = diff(WWdxdy, y);
\begin{align*}
\text{WWddxxy} & = \text{diff}(\text{WWdxdy}, x) \\
\text{WWddxdyy} & = \text{diff}(\text{WWddxdy}, y)
\end{align*}

\[\text{gradtofour} = \text{WWddddx} + 2 \cdot \text{WWddxddy} + \text{WWddddy};\]

\[\text{keNum} = D \cdot \int \left( \int \text{gradtofour}, y, 0, b/2 \right), 0, a/2;\]
\[x = 0.03175; \quad y = 0.0254;\]
\[\text{EGV} = \text{eval}(\text{WW1}) + \text{eval}(\text{WW2});\]
\[\text{keDen} = \text{EGV}^2;\]
\[\text{ke} = \frac{\text{keNum}}{\text{keDen}};\]

\subsection*{5.2 Appendix B – Passive Simulink Diagrams}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{passive_simulink_diagram}
\caption{Passive Suspension Model 1}
\end{figure}
5.3 Appendix C – Active Suspension Models

Figure 45 – Active Suspension Reference - No Controller
5.4 Appendix D – Controller Tests, MATLAB m-files

pid_test.m

% Batch simulation for PID 1mass,1mode
% 50 seconds @ time step=1e-4=.0001
% No disturbance, reference signal=.01

j=1;

p=1;i=1;d=1;4
for p=1:100:100000;
    sim('PID_1mass1mode_ansys');
    samp=length(ce(:,1));
    % Controller Effort & Performance
    contp(j,1)=ce(samp,2);
    contp(j,2)=cp(samp,2);
    j=j+1;
end

j=1;
p=1;i=1;d=1;
for i=1:100:100000;
    sim('PID_1mass1mode_ansys');
    samp=length(ce(:,1));
    % Controller Effort & Performance
    conti(j,1)=ce(samp,2);
    conti(j,2)=cp(samp,2);
    j=j+1;
end

j=1;
p=1;i=1;d=1;
for d=1:100:100000;
    sim('PID_1mass1mode_ansys');
    samp=length(ce(:,1));
    % Controller Effort & Performance
    contd(j,1)=ce(samp,2);
    contd(j,2)=cp(samp,2);
    j=j+1;
end
lqr_test.m

Tsim=1;
timestep=1e-4;

%LQR gain test
sim('LQR_1mass');
samp=length(lqrCE(:,1));

%no disturbance
j=1;
xd=1;
A=6.3e-3;
w=1000*2*pi;

for rho=1:10:1000;
sim('LQR_1mass');
eff(j,1)=lqrCE(samp,2);
per(j,2)=lqrCP(samp,2);
j=j+1;
end

batchparameters_RA_2mass.m

alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
A=6.3e-3; w=1000*2*pi; n=1;
sim('RA_2mass');
samp=length(CE(:,1));

%For no disturbance d(t)
%For Length = 100s, timestep tao=1e-3
%For each sgn, sat, tanh
%Desired, where xd=.01

for j=1:8;
v=1;
if j==1;
  alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
  for i=.1:.1:3;
    alpha=i; %Sliding Mode Controller Gain
    sim('RA_2mass')
    %Controller Effort & Performance
    conttot(1,v)=ARACE(samp,2);
    conttot(2,v)=ARACP(samp,2);
    v=v+1;
  end
end
v=1;
if j==2;
    alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
    for i=.1:.1:3;
        beta=i; %Sliding Mode Controller Gain
        sim('RA_2mass')
        %Controller Effort & Performance
        conttot(3,v)=ARACE(samp,2);
        conttot(4,v)=ARACP(samp,2);
        v=v+1;
    end
end

v=1;
if j==3;
    alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
    for i=.1:.1:3;
        L1=i; %Sliding Mode Controller Gain
        sim('RA_2mass')
        %Controller Effort & Performance
        conttot(5,v)=ARACE(samp,2);
        conttot(6,v)=ARACP(samp,2);
        v=v+1;
    end
end

v=1;
if j==4;
    alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
    for i=.1:.1:3;
        L2=i; %Sliding Mode Controller Gain
        sim('RA_2mass')
        %Controller Effort & Performance
        conttot(7,v)=ARACE(samp,2);
        conttot(8,v)=ARACP(samp,2);
        v=v+1;
    end
end

v=1;
if j==5;
    alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
    for i=.1:.1:3;
        L3=i; %Sliding Mode Controller Gain
sim('robadapt_2mass_1_19_WORKS_dist')
%Controller Effort & Performance
conttot(9,v)=ARACE(samp,2);
conttot(10,v)=ARACP(samp,2);
v=v+1;
end
end

v=1;
if j==6;
alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
for i=1:.1:3;
g1=i; %Sliding Mode Controller Gain
sim('RA_2mass')
%Controller Effort & Performance
conttot(11,v)=ARACE(samp,2);
conttot(12,v)=ARACP(samp,2);
v=v+1;
end
end

v=1;
if j==7;
alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
for i=1:.1:3;
g2=i; %Sliding Mode Controller Gain
sim('RA_2mass')
%Controller Effort & Performance
conttot(13,v)=ARACE(samp,2);
conttot(14,v)=ARACP(samp,2);
v=v+1;
end
end

v=1;
if j==8;
alpha=1; beta=1; g1=1; g2=1; g3=1; L1=1; L2=1; L3=1;
for i=1:.1:3;
g3=i; %Sliding Mode Controller Gain
sim('RA_2mass')
%Controller Effort & Performance
conttot(15,v)=ARACE(samp,2);
conttot(16,v)=ARACP(samp,2);
v=v+1;
end
end
end %ends the j=1:8 Loop

alpha=10; beta=10; g1=1; g2=1; g3=1; L1=1; L2=1; L3=10;
A=6.3e-3; w=1000*2*pi;

v=1;
for n=0.5:0.05:1.5;
    sim('RA_2mass');
    %Controller Effort & Performance
    contad(1,v)=ARACE(samp,2);
    contad(2,v)=ARACP(samp,2);
    v=v+1;
end

v=1;
A=6.3e-3; w=1000*2*pi;
for A=.001:.001:.01;
    sim('RA_2mass');
    %Controller Effort & Performance
    controb(1,v)=ARACE(samp,2);
    controb(2,v)=ARACP(samp,2);
    v=v+1;
end

v=1;
A=6.3e-3; w=1000*2*pi;
for whz=400:200:2000;
    w=whz*2*pi;
    sim('RA_2mass');
    %Controller Effort & Performance
    controb(3,v)=ARACE(samp,2);
    controb(4,v)=ARACP(samp,2);
    v=v+1;
end

compareall.m

Tsim=100 %time of simulation [seconds]
timestep=1e-4; %timestep, tao
%P=1;l=1;D=1;
sim('RA_1mass');
samp=length(RACE(:,1));
timestep=1e-5;
sim('PID_1mass');
sampi=length(pidCE(:,1));
for j=1:2; % 1 mass plant for each controller
  if j==1; % no disturbance, known parameters
    n=1; A=6.3e-3; w=1000*2*pi;
    xd=0; % vibration suppression
    % Set Optimal Values
    P=1e3; I=1e6; D=1; % PID Values
    timestep=1e-5;
    sim('PID_1mass');
    for i=10:10:sampi;
      contot(1,1)=pidCE(i,2)+contot(1,1);
      contot(1,2)=pidCP(i,2)+contot(1,2);
    end
    timestep=1e-4;
    rho=1e-6; % LQR & Observer values
    sim('LQR_1mass')
    contot(2,1)=lqrCE(samp,2);
    contot(2,2)=lqrCP(samp,2);
    sim('lqr_observer')
    contot(3,1)=obsCE(samp,2);
    contot(3,2)=obsCP(samp,2);
    a1=100;a2=100;L=200; % RA Values
    sim('RA_1mass')
    contott(4,1)=RACE(samp,2);
    contott(4,2)=RACP(samp,2);
    % memgain=1;alpha=10;beta=10;L1=1;L2=1;L3=20; % advanced RA
    % sim('RA_2mass')
    % contott(5,1)=ARACE(samp,2);
    % contott(5,2)=ARACP(samp,2);
  end % ends "if" statement
  if j==2; % 1 mass plant for each controller
    n=1.5; A=6.3e-3; w=1000*2*pi;
    xd=0; % tracking problem
    % Set Optimal Values
    P=1e3; I=1e6; D=1; % PID Values
    timestep=1e-5;
    sim('PID_1mass');
    for i=10:10:sampi;
      contot(1,1)=pidCE(i,2)+contot(1,1);
      contot(1,2)=pidCP(i,2)+contot(1,2);
    end
    timestep=1e-4;
    rho=1e-6; % LQR & Observer values
  end % ends "if" statement
if j==3; %1mass plant for each controller
    %disturbance present, known parameters
    n=1.25; A=0; w=1000*2*pi;
    xd=.01;
    %Set Optimal Values
    P=1e3; I=1e6; D=1; %PID Values
    timestep=1e-5;
    sim('PID_1mass');
    for i=10:10:sampi;
        contot(1,1)=pidCE(i,2)+contot(1,1);
        contot(1,2)=pidCP(i,2)+contot(1,2);
    end
    timestep=1e-4;
    rho=1e-6; %LQR & Observer values
    sim('LQR_1mass')
    contot(12,1)=lqrCE(samp,2);
    contot(12,2)=lqrCP(samp,2);
    sim('lqr_observer')
    contot(13,1)=obsCE(samp,2);
    contot(13,2)=obsCP(samp,2);
    sim('RA_1mass')
    contott(14,1)=RACE(samp,2);
    contott(14,2)=RACP(samp,2);
    %sim('RA_2mass')
    %contott(15,1)=ARACE(samp,2);
    %contott(15,2)=ARACP(samp,2);
end

if j==4; %2mass plant for each controller
    %disturbance present, known parameters
    n=1; A=6.3e-3; w=1000*2*pi;
    xd=.01;
% Set Optimal Values
P=1e3; I=1e6; D=1; % PID Values

timestep=1e-5;
sim('PID_1mass');
for i=10:10:sampi;
    contot(1,1)=pidCE(i,2)+contot(1,1);
    contot(1,2)=pidCP(i,2)+contot(1,2);
end

timestep=1e-4;
rho=1e-6; % LQR & Observer values
sim('LQR_1mass')
    contot(17,1)=lqrCE(samp,2);
    contot(17,2)=lqrCP(samp,2);
sim('lqr_observer')
    contot(18,1)=obsCE(samp,2);
    contot(18,2)=obsCP(samp,2);
sim('RA_1mass')
    contott(19,1)=RACE(samp,2);
    contott(19,2)=RACP(samp,2);
end

if j==5; % 1 mass plant for each controller
    % disturbance present, known parameters
    n=1.25 ;A=6.3e-3; w=1000*2*pi;
    xd=.01;
    % Set Optimal Values
    P=1e3; I=1e6; D=1; % PID Values
    timestep=1e-5;
sim('PID_1mass');
    for i=10:10:sampi;
        contot(1,1)=pidCE(i,2)+contot(1,1);
        contot(1,2)=pidCP(i,2)+contot(1,2);
    end
    timestep=1e-4;
rho=1e-6; % LQR & Observer values
    sim('LQR_1mass')
        contot(22,1)=lqrCE(samp,2);
        contot(22,2)=lqrCP(samp,2);
sim('lqr_observer')
        contot(23,1)=obsCE(samp,2);
        contot(23,2)=obsCP(samp,2);
sim('RA_1mass')
        contott(24,1)=RACE(samp,2);
contott(24,2)=RACP(samp,2);
%sim('RA_2mass')
%contott(25,1)=ARACE(samp,2);
%contott(25,2)=ARACP(samp,2);
end

end %ends the j=1:5 Loop
6 References


