The Intelligent Combination of Input Shaping and PID Feedback Control

A Dissertation
Presented to
The Academic Faculty

by

John R. Huey

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Mechanical Engineering
Georgia Institute of Technology
August 2006

The Intelligent Combination of Input Shaping and PID Feedback Control

Approved by:

Dr. William Singhose, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Aldo Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Dr. David Frakes
4D Imaging, Inc.

Dr. Kok-Meng Lee
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Panagiotis Tsiotras
School of Aeronautical Engineering
Georgia Institute of Technology

Date Approved: July 3, 2006

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. William Singhose for all of his help and guidance throughout my five years in graduate school. Thanks also to my other committee members Dr. Aldo Ferri, Dr. David Frakes, Dr. Kok-Meng Lee and Dr. Panagiotis Tsiotras. I would also like to extend a special thanks to Dr. Aldo Ferri, who first introduced me to the idea of attending graduate school.

I would like to thank my parents, Marty and Gail Huey, for their many years of love and support. And, of course, I would like to thank my wife, Randee Huey, for being tremendously supportive and a great cheerleader in my life and academic pursuits.

Finally, I would like to thank my Lord and Savior Jesus Christ for His unending loving-kindness.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Input shaping and Proportional-Integral-Derivative (PID) feedback control are simple, easy-to-implement and generally low cost control strategies. Considering this, it is remarkable that they are also very effective control techniques. In fact, a majority of the world's feedback controllers utilize PID (or the subset PD) control. In addition, input shaping has seen significant use on real-world machines such as cranes, micro-mills, coordinate measuring machines, computer disc drive manufacturing machines, spacecraft, etc.

However, despite similarities in effectiveness and ease of implementation, input shaping and PID feedback control are fundamentally different strategies. Input shaping is an anticipatory control scheme capable of enabling quick, low-vibration motions. PID feedback control is reactive in nature, and it is primarily required to deal with problems such as modeling errors, disturbances and nonlinearities.

Given their effectiveness and practicality, as well as the fact that they address important and complimentary control issues, it would be advantageous to combine these two control strategies. The result would still be practical and effective, yet would now address a range of system phenomenon beyond that which is capable by either of the individual control techniques. However, there is a definite gap in the state-of-the-art technology for combining these techniques. For example, little research has addressed the intelligent combination of traditional, outside-the-loop input shaping and PID feedback control. In addition, only a few researchers have attempted to place input shaping filters within feedback loops.

This research studies the intelligent combination of input shaping and PID feedback control by developing a concurrent design procedure for outside-the-loop input shaping/PID feedback combinations and by analyzing the effect of placing input shaping filters within feedback loops.

CHAPTER I

INTRODUCTION

1.1 Problem Statement

The control of flexible systems is an immense field of research. There are a variety of reasons why mechanical systems are designed and built to be flexible. For example, many mechanical systems need to be lightweight. Lightweight systems can be moved faster and/or with less energy than heavier mechanical systems. Unfortunately, making a mechanical system lightweight usually means that it will also be flexible. And, if speed is a primary goal, then vibration control will be a necessity. Flexible systems moving at high accelerations and velocities will, generally vibrate. This vibration can cause a variety of problems including positioning errors, slow overall move times (if vibration must naturally damp out), and system damage.

The three primary methods for limiting vibrations on flexible, mechanical systems are to intelligently choose motion commands, to utilize some form of feedback control or to move so slowly that the flexible dynamics are not excited. Given that moving slowly is undesirable for many reasons, this research seeks to study the combination of command generation and feedback control. Particularly, this research will focus on intelligently combining input shaping and Proportional-Integral-Derivative (PID) control. However, the lessons learned here are applicable to the more general areas of command generation (or command shaping) and feedback control.

The combination of input shaping and PID feedback control has been studied and implemented in many controls applications. However, there is a deficiency in the state-of-the-art technology. First, while outside-the-loop input shaping and PID feedback controllers have been extensively combined, little research has addressed the intelligent combination of these two control techniques. That is, the PID gains and the input shaper parameters are usually derived separately, and the two techniques are then combined without accounting for the combined dynamic effects. Figure 1.1 depicts this sequential design process. Figure 1.2 depicts the design process that this research will focus on, where the command generator and feedback system are designed together according to the given plant and desired system behavior.

Secondly, the vast majority of input shaping/PID feedback combinations have utilized the input shaper outside of any feedback loop. Only a few researchers have

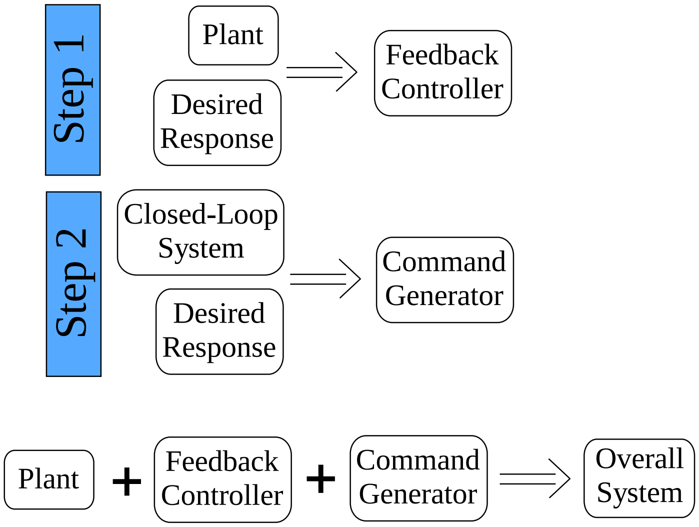


Figure 1.1: Sequential Design Process.

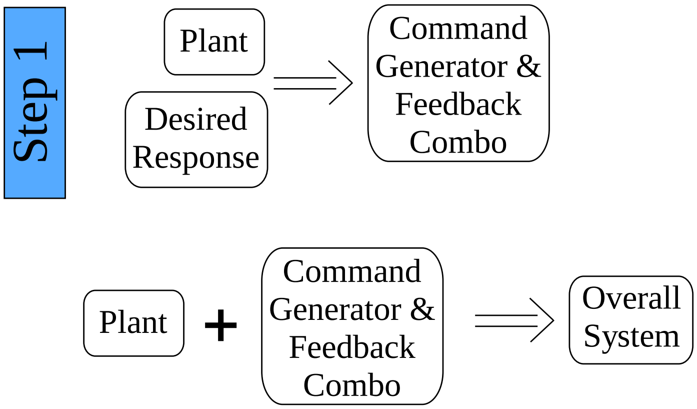

Step 1
Plant
Desired Response
Command Generator & Feedback Combo
Plant
+
Command Generator & Feedback Combo
Overall System

Figure 1.2: Concurrent Design Process.

attempted to place input shaping filters within feedback loops. This thesis will provide a framework for placing input shapers within feedback loops. Stability issues will be studied first, followed by an investigation into the useful applications for this type of controller.

1.2 Command Generation and Command Shaping for Vibration Reduction

Command generation is a technique that specifically designs unique reference commands for a given system. The unique reference command is chosen based on its ability to drive the system according to a set of performance constraints. One example of command generation is the use of S-curves to drive flexible systems.

Command shaping is a similar process by which a desired reference command is modified so as to improve the performance of a given system. For instance, the reference command might be altered so as to reduce the residual vibration which typically would result from an unmodified reference command. One example of command shaping is the use of low-pass filters.

Depending on the application, some of these command shaping techniques will be more useful than others. Many command shaping techniques have little robustness to modeling errors. In addition, many are not applicable in real time, requiring pre-computation of command functions.

1.3 Input Shaping

One very useful form of command shaping is input shaping. Input shaping is applicable in real time, and input shapers can be designed to have any desirable robustness level. Input shaping is designed to reduce, or eliminate, command-induced system vibration. A desired reference command given to a flexible system will, in general, result in residual vibration. However, if the system's natural frequency and damping

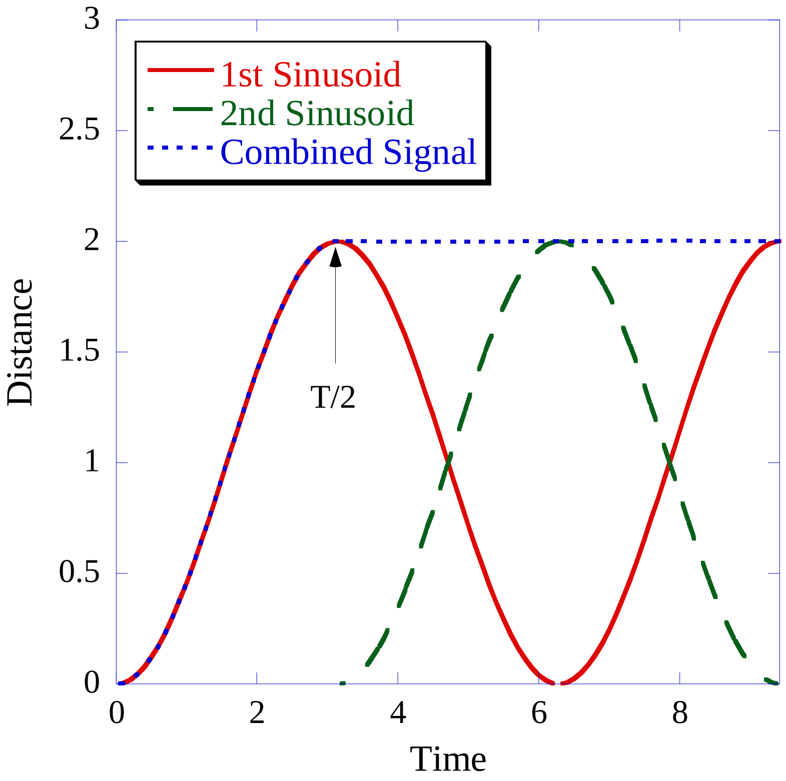


Figure 1.3: Destructive Interference.

ratio are known, then any reference command can be altered so as to produce little or no residual vibration in the system response.

Input shaping's ability to cancel vibration can be viewed as destructive interference of sinusoidal waves. If two sinusoids of the same magnitude, same frequency and correct phase shift between them are added together, the resulting combination will have no oscillations. This effect can be seen in Figure 1.3. This concept can be extended to the vibration reduction of flexible systems. If a flexible system with a constant natural frequency is given two equal inputs correctly spaced in time, then the vibration resulting from each input will add destructively to yield zero residual vibration. Note that the earliest time at which the second input, or sinusoid, can be added is at one half the natural period of the system. This time is labeled $T/2$ in Figure 1.3.

Input shaping operates by creating a sequence of delayed impulses that, if given

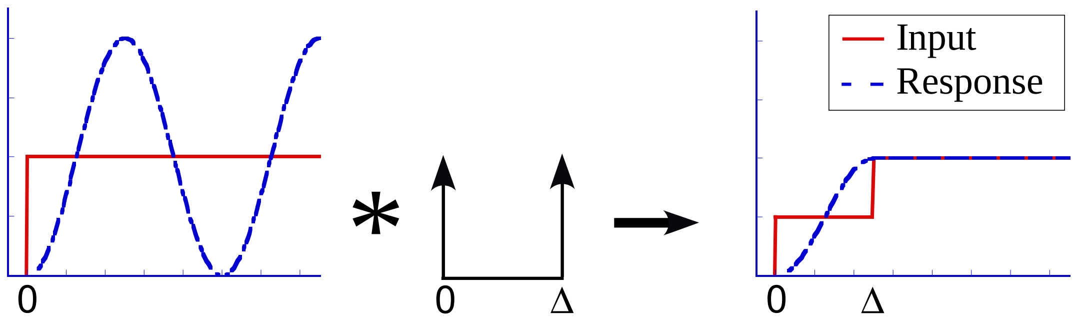


Figure 1.4: Input Shaping Process.

to a flexible system, will cause the destructive interference shown in Figure 1.3. The sequence of impulses, known as the input shaper, is then convolved with a desired reference command to produce a new, modified command that can be used to drive the system. As shown in Figure 1.4, this modified command will cause the system to move with no residual vibration. The sequence of impulses (in particular the impulse times and amplitudes) is chosen such that, when the modified command is applied to the system, certain performance constraints are met. These performance constraints can include the system's desired residual vibration amplitude, robustness to modeling errors, and command rise time, among others.

1.3.1 Alternative Perspectives

There are several other ways to describe input shaping. One is to say that the impulse sequence filters out the system's natural frequency from the reference command. This is done so that the natural frequency is not excited in the system when the new, modified command is applied to it. Another perspective on input shaping comes from analyzing the poles and zeros of a system. For example, Figure 1.4 shows a two-impulse shaper called a "Zero Vibration" (ZV) shaper [65, 85]. The ZV shaper can be expressed in the Laplace domain by the following equation:

$$ZV(s) = A_1 + A_2 e^{-s\frac{T}{2}} \tag{1.1}$$

This equation constitutes an impulse of magnitude A_1 occurring at time $t = 0$ and a second impulse of magnitude A_2 delayed by $\frac{T}{2}$ seconds. If T is the damped natural period of the system, then shaping a reference command with this ZV shaper will eliminate the vibratory mode associated with the period T. Note that the reference command must eventually reach a constant, steady-state value. In addition, the vibration is only guaranteed to be eliminated by some time after this steady state has been reached. Input shaping cannot eliminate vibration caused by continuously changing reference commands like sinusoids. Solving for the poles and zeros of this ZV shaper yields:

$$zeros \rightarrow s = -\frac{2}{T} ln\left(\frac{A_2}{A_1}\right) \pm j\omega_d(2k+1) \qquad where \;\; k = 0, 1, 2, ... \tag{1.2}$$

$$poles \rightarrow s = -\infty \pm j\omega \tag{1.3}$$

In these two equations, ω_d is the natural frequency associated with the damped period, T. The ω left undefined in the “poles” equation is intended to indicate that any value would still yield an open-loop pole.

As can be seen from (1.2), an input shaper creates an infinite column of zeros. If an input shaper is correctly designed, then one pair of these shaper zeros will cancel the oscillatory poles of a flexible system. Figure 1.5 shows the poles of a second-order system being canceled by a two-impulse shaper similar to the one shown in Figure 1.4. The open-loop poles, while not relevant here, will become important when shapers are included within feedback loops.

As can be seen from Figure 1.4, input shaping does come with one obvious cost. Convolving a reference command with an input shaper adds a delay to the rise time of the system. However, in many practical situations, the delay added to the system move is more than made up for by the drastic reduction in settling time.

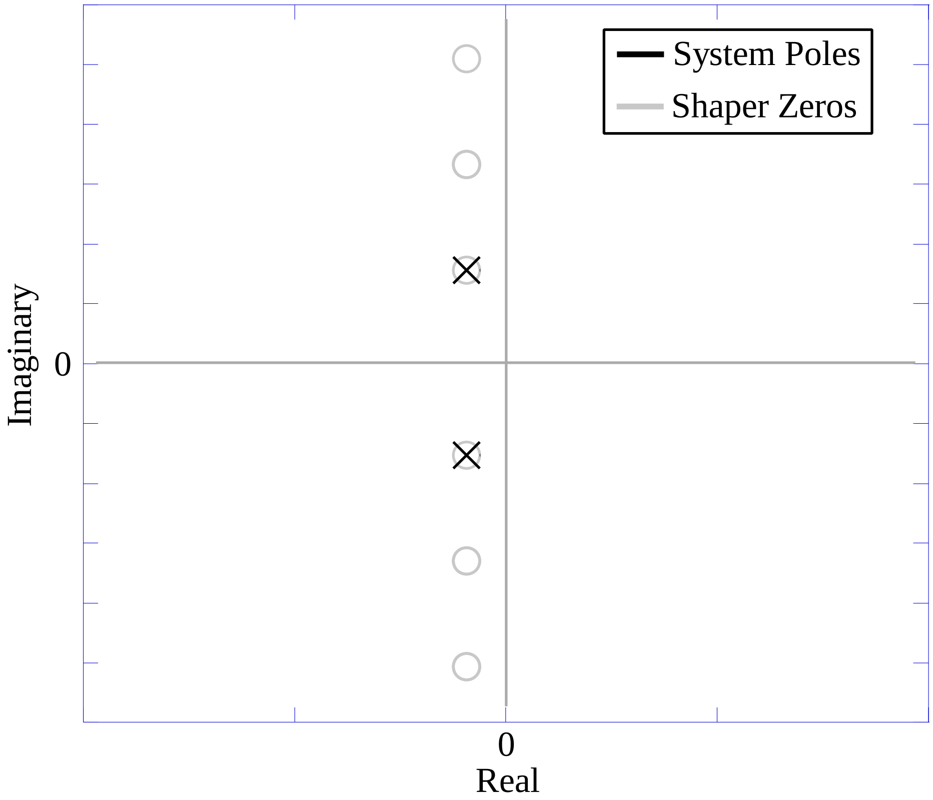


Figure 1.5: Pole/Zero Cancellation.

1.3.2 Brief History of Input Shaping

Originally named “Posicast Control”, the initial development of input shaping is largely credited to O.J.M. Smith [85] during the late 1950’s. However, there seems to have been one notable precursor to “Posicast Control”. In the early 1950’s, John Calvert developed a time-delay based vibration filter named “Signal Component Control” [4,15]. However, his solution did not contain the convenient closed-form description offered by Smith.

Since this initial work, there have been many developments in the area of input shaping control. For example, input shapers have been developed that are robust to natural frequency modeling errors, the first of which was called the Zero Vibration and Derivative (ZVD) shaper [65]. Input shapers for multi-mode systems have also been designed [23, 58, 65, 78, 93, 98]. In addition, some work has addressed the use of input shaping on nonlinear systems [17,34,81]. Other examples include systems with Coulomb friction [40,41], backlash [38], and on-off thrusters [44,62,66,68,77]. Input shaping has also been applied to systems with varying parameters [45,46,57].

Because the simple forms of input shaping are easy to implement, they have been applied to many real-world systems with great success. Some of the notable real-world applications include cranes [43,64,70,71,87–89], disk drive manufacturing machines [75], coordinate measurement machines [24,63,73,74], micro-milling machines [13], flexible spacecraft [17,76,79,97], telecommunications [22], long-reach manipulators [36], and tele-robotic arms [20].

Traditionally, input shaping is an open-loop control strategy developed for linear (or near-linear) systems. Input shaping can be considered as a predictive control scheme. That is, it uses knowledge of the system to re-shape reference commands such that undesired system behavior does not occur. Therefore, input shaping allows for quick, low-vibration motions. Unfortunately, if the unexpected occurs (i.e. modeling errors, disturbances, nonlinearities, etc.), the effectiveness of traditional input shaping can be degraded. Fortunately, for the areas of modeling errors and nonlinearities, a significant amount of research has been done that allows input shaping to work well under these conditions. However, as with any open-loop controller, traditional input shaping can do nothing to address disturbances.

1.3.3 Important Input Shapers

This section will review some of the most well known input shapers. Each of these input shapers is used within the research described by this thesis. The equations that detail the impulse times and impulse amplitudes will be written in matrix form. The first row will indicate the time by which each impulse is delayed. The second row indicates the amplitude of the impulse. Each impulse is completely defined by one column: a time delay and amplitude value. A generic input shaper depicted in matrix form is shown in the following equation. Note that the first impulse (A_1) will occur at time $t_1 = 0$ for all input shapers described in this section.

$$\begin{bmatrix} 0 & t_2 & t_3 & ... \\ A_1 & A_2 & A_3 & ... \end{bmatrix} \tag{1.4}$$

1.3.3.1 Zero Vibration Shaper (ZV)

The Zero Vibration (ZV) shaper is one of the oldest and simplest input shapers [65,85]. This input shaper is designed for one main purpose – to filter an incoming signal such that a system driven by the new, shaped signal will have no vibration arising from the frequency filtered out by the ZV shaper. A ZV shaper has two impulses. The equations describing the ZV shaper are:

$$\begin{bmatrix} 0 & t_2 \\ A_1 & A_2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{\pi}{\omega_d} \\ \frac{\gamma}{1+\gamma} & \frac{1}{1+\gamma} \end{bmatrix} \tag{1.5}$$

where:

$$\gamma = e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{1.6}$$

In these equations, ζ and ω_d are the damping ratio and damped natural frequency (respectively) of the oscillatory mode addressed by the *ZV* shaper.

1.3.3.2 Zero Vibration and Derivative Shaper (ZVD)

The Zero Vibration and Derivative shaper (ZVD) is similar to the ZV shaper in that it is designed to yield zero vibration at some modeled frequency [65]. However, the ZVD shaper has an added constraint not seen in the ZV shaper derivation. The ZVD shaper has added robustness to modeling errors. This is achieved by not only forcing the vibration to be zero at the modeled frequency, but by also forcing the derivative of the vibration amplitude with respect to frequency to be zero. The cost for the added robustness of the ZV shaper is that the ZVD shaper requires twice the amount of time. The equations for the ZVD shaper are:

$$\begin{bmatrix} 0 & t_2 & t_3 \\ A_1 & A_2 & A_3 \end{bmatrix} = \begin{bmatrix} 0 & \frac{\pi}{\omega_d} & \frac{2\pi}{\omega_d} \\ \frac{\gamma^2}{\gamma^2+2\gamma+1} & \frac{2\gamma}{\gamma^2+2\gamma+1} & \frac{1}{\gamma^2+2\gamma+1} \end{bmatrix} \tag{1.7}$$

where:

$$\gamma = e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{1.8}$$

1.3.3.3 Unity Magnitude Zero Vibration Shaper (UMZV)

Unlike the ZVD shaper, the Unity Magnitude Zero Vibration shaper (UMZV) prioritizes speed and gives up robustness to modeling errors [75]. The UMZV shaper is quicker than the ZV shaper, but has less robustness. The equations for the UMZV shaper can only be shown in an analytical form when the shaper is designed for an undamped, oscillatory mode. If $\zeta = 0$, then the impulse times and amplitudes are:

$$\begin{bmatrix} 0 & t_2 & t_3 \\ A_1 & A_2 & A_3 \end{bmatrix} = \begin{bmatrix} 0 & \frac{\pi}{3\omega_n} & \frac{2\pi}{3\omega_n} \\ 1 & -1 & 1 \end{bmatrix} \tag{1.9}$$

1.3.3.4 Extra-Insensitive Shaper (EI)

The Extra-Insensitive shaper (EI) increases the robustness of the ZVD shaper by keeping the derivative constraint at the modeled frequency but relaxing the zero-vibration constraint at the modeled frequency [72]. That is, the EI shaper actually requires some non-zero level of vibration when the oscillatory system being aided by the input shaper is perfectly modeled. This non-zero level of vibration is variable, and is labeled as V in the following equations. Again, the equations for the EI shaper can only be shown in an analytical form when the shaper is designed for an undamped, oscillatory mode. If $\zeta = 0$, then the impulse times and amplitudes are:

$$\begin{bmatrix} 0 & t_2 & t_3 \\ A_1 & A_2 & A_3 \end{bmatrix} = \begin{bmatrix} 0 & \frac{\pi}{\omega_n} & \frac{2\pi}{\omega_n} \\ \frac{1+V}{4} & \frac{1-V}{2} & \frac{1+V}{4} \end{bmatrix} \tag{1.10}$$

1.3.3.5 Sensitivity Plots

The robustness of each of the input shapers just described can be pictorially viewed on a sensitivity plot. An input shaper sensitivity plot shows the amount of residual vibration an underdamped, single-mode system will have if given a command filtered by the input shaper in question. The amplitude of the residual vibration caused by the input shaped signal is plotted as a percentage of the amplitude of the residual vibration that would be caused by the original, unshaped signal. This percent residual vibration is plotted as a function of the ratio between the actual damped, natural frequency of the single-mode system (ω_a) and the frequency used to design the input shaper (ω_m).

Figure 1.6 shows the sensitivity plots of each of the input shapers described so far in this section. Note that ZV, ZVD and UMZV shapers have zero vibration when

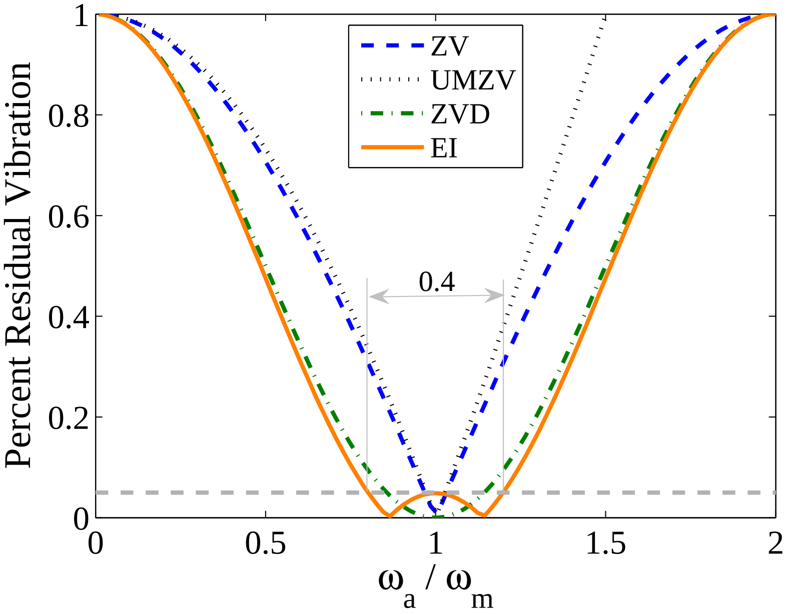


Figure 1.6: Sensitivity Plot Comparison.

$\omega_m = \omega_a$. However, the EI shaper intentionally has a non-zero residual vibration when $\omega_m = \omega_a$. This is done so that the EI shaper will be more insensitive to modeling errors. A shaper's insensitivity is measured by the width of the continuous spectrum of normalized frequency resulting in less than a specified percent residual vibration. For example, Figure 1.6 shows the 5% insensitivity of an EI shaper designed to have that amount of vibration when $\omega_m = \omega_a$. Note from this figure, that the EI shaper has the highest 5% insensitivity, followed by the ZVD, the ZV, and then the UMZV shaper.

1.3.3.6 Specified-Negative-Amplitude Shaper (SNA)

One final input shaper worth noting here is the Specified-Negative-Amplitude shaper (SNA) [69]. The ZV, ZVD and EI shapers all have positive amplitude impulses. Of the shapers previously described, only the UMZV allows for some impulses to have negative amplitude. The advantage gained by this allowance is speed, as the UMZV is the fastest of the previously described input shapers. The disadvantage is a lack of robustness and the fact that input shapers with negative amplitudes can actually magnify high frequency vibrations. This is indicated in Figure 1.6 where the slope of the sensitivity curve for the UMZV shaper is non-zero at $\frac{\omega_a}{\omega_m} = 1.5$. Note that the sensitivity curves for the other three input shapers flatten out near $\frac{\omega_a}{\omega_m} = 2$, peaking at a percent residual vibration of one.

SNA shapers were designed to create a continuous spectrum of input shapers that easily balance the tradeoff between speed and robustness/high mode excitation. All SNA shapers are of the same form: positive impulse, negative impulse, positive impulse. They are designed by specifying the negativity of the second impulse (denoted by "b" in the following equations), which then also determines the amplitudes of both positive impulses and the impulse times t_2 and t_3. It can be shown that the boundaries of the SNA spectrum are the UMZV shaper (when the negativity is $b = -1$)

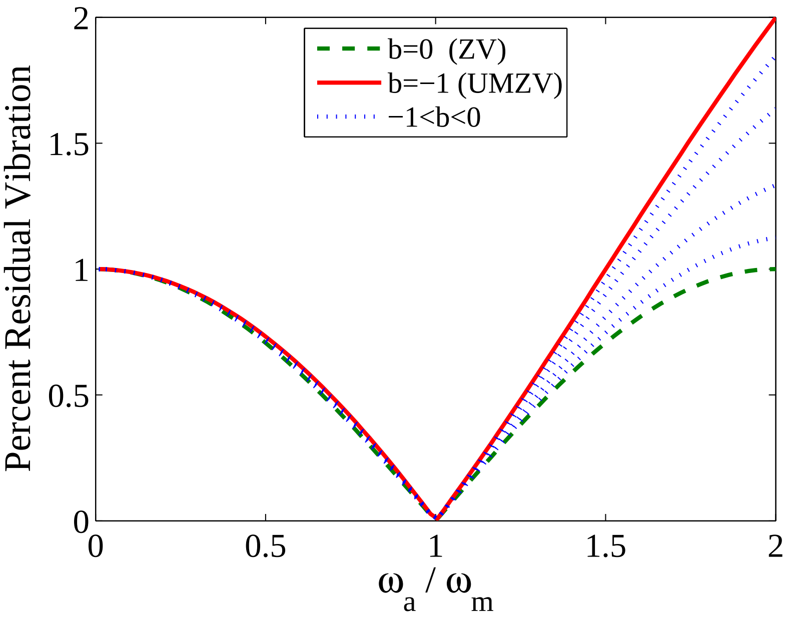


Figure 1.7: SNA Shaper Sensitivity Plots.

and the ZV shaper (when the negativity is $b = 0$).

The equations for the SNA shaper can only be shown in an analytical form when the shaper is designed for an undamped, oscillatory mode. If $\zeta = 0$, then the impulse times and amplitudes are:

$$\begin{bmatrix} 0 & t_2 & t_3 \\ A_1 & A_2 & A_3 \end{bmatrix} = \begin{bmatrix} 0 & t_2 & t_3 \\ \frac{1-b}{2} & b & \frac{1-b}{2} \end{bmatrix} \tag{1.11}$$

where:

$$t_2 = \frac{1}{\omega_n}\cos^{-1}\left(\frac{-b}{1-b}\right) \qquad t_3 = \frac{1}{\omega_n}\cos^{-1}\left(\frac{2b^2}{(1-b)^2} - 1\right) \tag{1.12}$$

Figure 1.7 shows the spectrum of SNA shapers via a sensitivity plot. Note that the UMZV and ZV shapers from Figure 1.6 are shown here as the boundary shapers for the SNA spectrum.

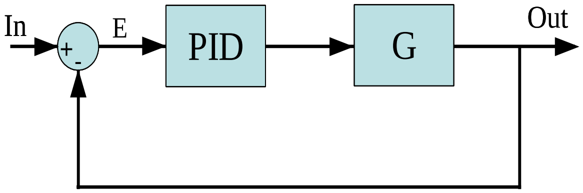


Figure 1.8: PID Feedback Controller.

1.4 PID Feedback Control

Feedback control is one of the oldest, most successful, and thoroughly researched control strategies. Even though an extraordinary number of feedback controllers have been developed, by far the most common form of feedback control is Proportional-Integral-Derivative (PID) control [14,55,92]. For example, a vast number of industrial machines utilize some form of PID control, often times with the integral term being set very low or removed altogether. This leads to the important subset of PD control. Some of the main applications of feedback control include rejecting disturbances, eliminating steady-state error and handling non-zero initial conditions. Unfortunately, outside-the-loop input shaping cannot specifically address any of these issues.

Figure 1.8 shows the basic form of PID feedback control. Here, the reference signal (labeled “In”) is compared to the output signal (labeled “Out”) to create an error signal (labeled “E”). The actuator signal sent to the plant (“G”) is a combination of three possible signals generated by the PID controller block. The first is an actuator effort proportional to the error signal. The second is an actuator effort proportional to the derivative of the error signal. The third is an actuator effort proportional to the integral of the error signal. Reactive in nature, PID control responds to a measured error so as to eliminate it. By doing this, problems such as modeling errors and disturbances are readily dealt with – although, only after they cause some error. In addition, simple PD control is capable of stabilizing some nonlinear systems.

However, because PID feedback control is inherently reactive, there is often a speed (as measured by rise time and settling time) and/or vibration reduction limitation

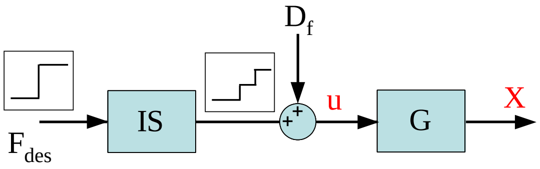


Figure 1.9: Outside-the-Loop Input Shaping Controller.

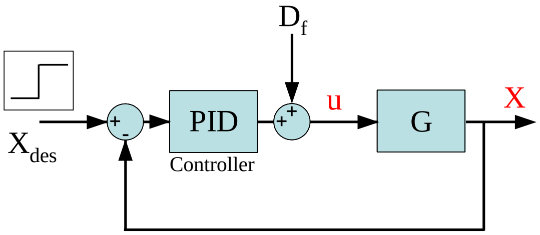


Figure 1.10: PID Feedback Controller.

when the system responds to reference commands. For instance, most systems have actuator limits which prevent large PID gains from being used in realistic scenarios. Therefore, the speed and vibration characteristics of realistic PID systems cannot be arbitrarily chosen. Usually, the system can either be made fast or non-vibratory. PID control can rarely achieve both simultaneously if there are realistic actuator limits.

In addition to actuator limitations, stability requirements can also restrict the gains of a PID controller. Often, the PID controller cannot use high gains (that might enable a quick response), because these high gains lead to instability.

1.5 Comparison of Input Shaping and PID Feedback Control

A few simple simulations can be used to depict the typical tradeoffs in the choice between input shaping and PID feedback control. Figure 1.9 shows a typical input shaping control scheme with a step input and an unexpected disturbance. Figure 1.10 shows a typical PID feedback control scheme with the same input and disturbance. In both control schemes, G is simply a mass plant. The PID gains were chosen so as to have a fast rise time and minimal vibration. The input shaper was designed to

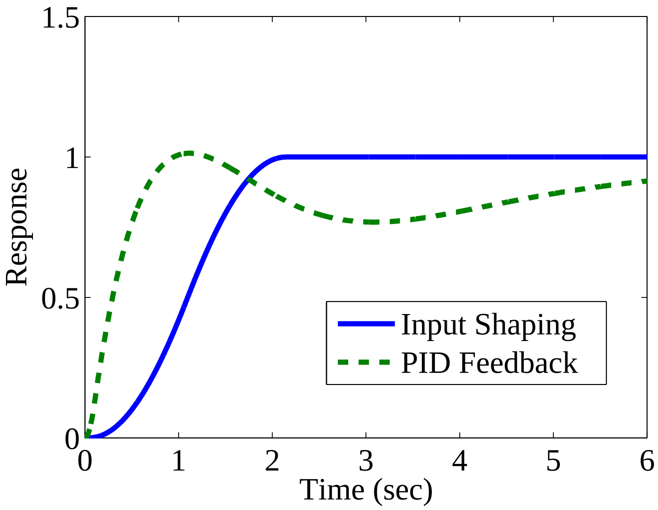


Figure 1.11: Response Comparison.

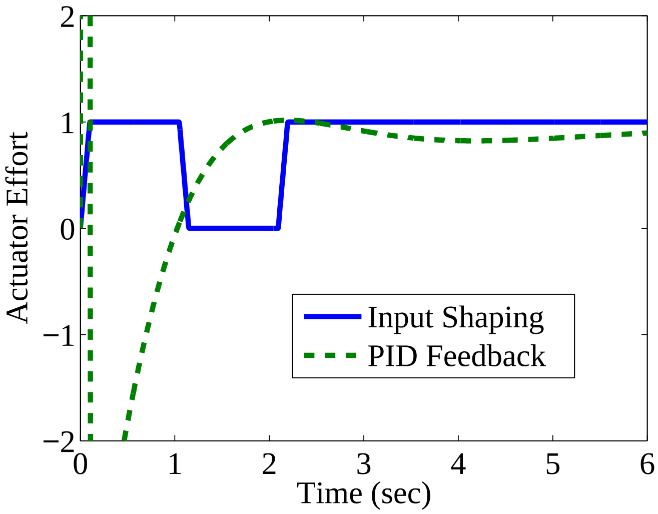


Figure 1.12: Required Actuator Effort Comparison.

change the reference command from a step to a bang-bang type command known to quickly move masses.

Figure 1.11 shows a unit step response (with $D_f = 0$) for each control scheme. The rise time of the PID control scheme is clearly shorter than that of the input shaped control scheme, although the overall settling time of the feedback control scheme is much longer. Figure 1.12 shows the actuator efforts (labeled "u" in Figures 1.9 and 1.10) required to generate the responses shown in Figure 1.11. For many

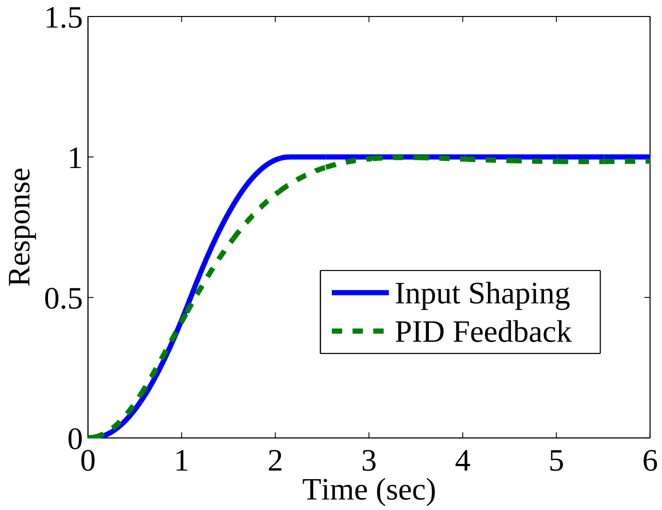


Figure 1.13: Bounded Response Comparison.

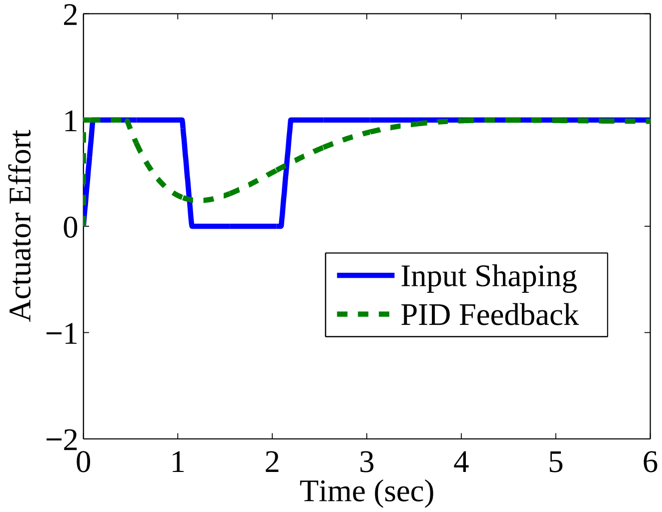


Figure 1.14: Bounded Actuator Effort Comparison.

practical situations, the actuator effort required by the PID controller is unrealistic, requiring a significant spike due to the differentiator combined with such an aggressive command. If the PID controller is forced within the same actuator bounds kept by the input shaping scheme, the PID scheme's response slows to that shown in Figure 1.13. Figure 1.14 shows the corresponding actuator requirements. It is quite clear from Figure 1.13 that the PID controller scheme is slower (in rise time and settling time) than the input shaping scheme. In fact, for the mass plant studied here, the best

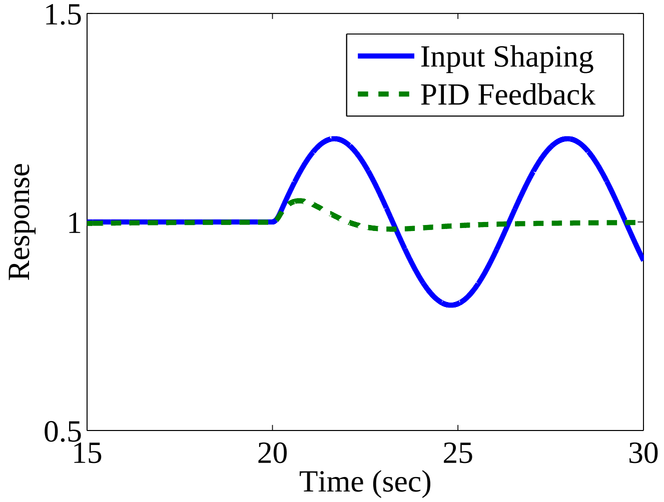


Figure 1.15: Disturbance Rejection Comparison.

the PID controller can do is to match the input shaping scheme's response. This is accomplished by setting the PID gains to infinity and limiting the actuator input to the 0-1 unit bounds used in these simulations. This would create the same bang-bang input generated by the input shaper. And, it is well known that a bang-bang command is the time optimal command for a mass.

While the previous figures seem to indicate input shaping's superiority, there are definitely applications where PID control is the better choice. Figure 1.15 shows a disturbance response for each of the two control schemes. Here, the disturbance force was a quick pulse signal intended to mimic an impulse. The PID controller quickly eliminates the disturbance's effect. The input shaping controller is helpless to solve this problem, due to its open-loop architecture.

These simulations clearly depict the strengths and weaknesses that were discussed earlier. Predictive in nature, traditional input shaping can yield quick, low-vibration motion. However, it cannot address certain issues such as disturbance rejection. On the other hand, PID control is reactive in nature, continuously monitoring the system's response. Therefore, it can address problems such as disturbance rejection. However, under realistic actuator limitations, it is most often slower in response to a reference input than is input shaping. In addition, because it is feedback control, it has the potential to cause instability. And finally, PID control requires integrators and differentiators, each of which can cause implementation problems in real systems.

1.6 Primary Research Question

In what ways can input shaping and PID feedback control be intelligently combined so as to produce a control system superior to current combinations or the individual implementations?

1.6.1 Definitions

This section will define some terminology that will be important throughout this dissertation. Figure 1.16 depicts each of the terms to be defined here. The abbreviation "OLIS" will stand for "Outside-the-Loop Input Shaping" and will refer to the controller shown in Figure 1.16 when the input shaping filter is placed outside of the feedback loop (position #1). Note that OLIS (using input shapers to pre-filter input signals to open or closed-loop systems) is the traditional form of input shaping, where the majority of research and applications occur. The abbreviation "CLSS" will stand for "Closed-Loop Signal Shaping" and will refer to the controller shown in Figure 1.16 when the input shaping filter is placed somewhere inside the feedback loop - for example in position #2. However, any control scheme with an input shaping filter somewhere within a feedback loop will be referred to as a CLSS controller. Note that although the filter within the feedback loop can be designed in the same manner as

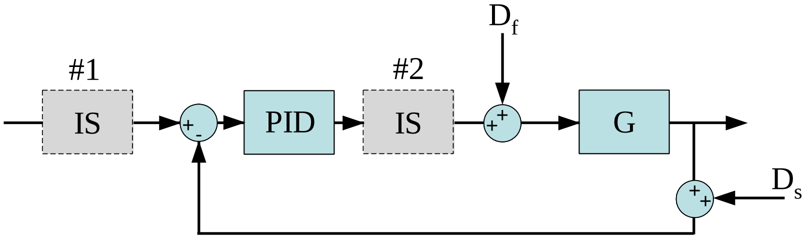


Figure 1.16: Description of Terminology.

a standard input shaper, its use within the feedback loop is outside of the traditional definition of “input shaping”. Therefore, from now on, controllers that contain input shaping filters within the feedback loop will be referred to as CLSS controllers.

There are also two disturbance signals which must be defined. “Force Disturbances” are disturbance forces which enter the block diagram just before the plant. That is, disturbance forces act directly on the plant. This signal is labeled D_f in Figure 1.16. “Sensor Disturbances” are disturbance signals which affect the feedback sensor. They are labeled D_s in Figure 1.16.

1.7 Literature Review

1.7.1 Outside-the-Loop Input Shaping and PID Feedback Control

Many researchers have, in some way, combined OLIS and feedback control. For example, Seth, *et al.*, used OLIS and feedback control to limit vibrations on a coordinate measuring machine [63]. Agostini, *et al.*, looked at the effects of combining OLIS and feedback control for use on a ship crane [1]. Magee, *et al.*, combined a time-delay pre-filter (OAT filter) with a feedback controller for use on robotic arms [49]. Dharne and Jayasuriya used an adaptive closed-loop controller to force the closed-loop dynamics to be well suited for OLIS even in the presence of modeling errors or unmodelled higher modes [9]. Finally, Chang and Park designed a robust feedback controller intended to make the closed-loop system exhibit dynamics well suited for use with OLIS [6].

Some researchers have even begun to look at the concurrent design of OLIS and feedback control parameters. Specific to PID feedback control, Kenison and Singhose developed an optimization routine to concurrently design OLIS and PD control parameters [30, 31]. Gopalakrishnan, Reddy and Singh studied the concurrent design of OLIS and PD control for several second-order systems [16]. Banerjee, Pedreiro and Gonzalez developed a nonlinear optimization routine to concurrently

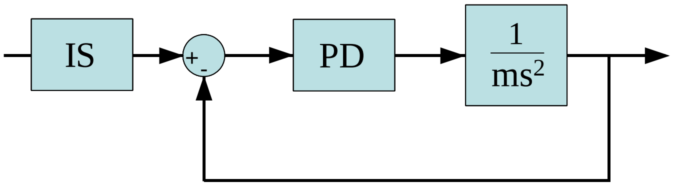


Figure 1.17: Outside-the-Loop Input Shaping and PD Control of a Mass.

design an outside-the-loop input shaping and PID feedback controller for a flexible spacecraft [3]. Going beyond PID control, Chang and Park advanced their earlier research [6] to create a concurrently designed input shaping and feedback combination for the same application [7]. Muenchhof and Singh used an optimization routine to concurrently design an input shaper and a state feedback controller [52].

The work completed by Kenison and Singhose was an excellent beginning to the concurrent design of outside-the-loop input shapers and PID feedback control [30,31]. They focused on the block diagram shown in Figure 1.17. Here, "IS" represents the input shaper and "PD" represents the proportional and derivative controller. They first optimized the PD gains for use without an input shaper and then concurrently optimized both the PD gains and the shaper parameters. As can be seen in Figure 1.18, for a unit step reference input and impulse disturbance, the input shaper/PD combination (labeled "CEF" for "Command-Enhanced Feedback") outperformed the PD control (labeled "Unshaped") alone in terms of overshoot, settling time, and disturbance rejection. Note that these responses are the same as those originally presented by Kenison and Singhose [30, 31].

1.7.1.1 Research Gap

While input shaping and PID feedback control are often combined, and even while some researchers have begun the study of concurrently designing outside-the-loop input shaping and PID feedback controllers, this area of research is still lacking a full, in-depth investigation as well as a general design method.

While they constitute excellent examples of combining OLIS with some form of

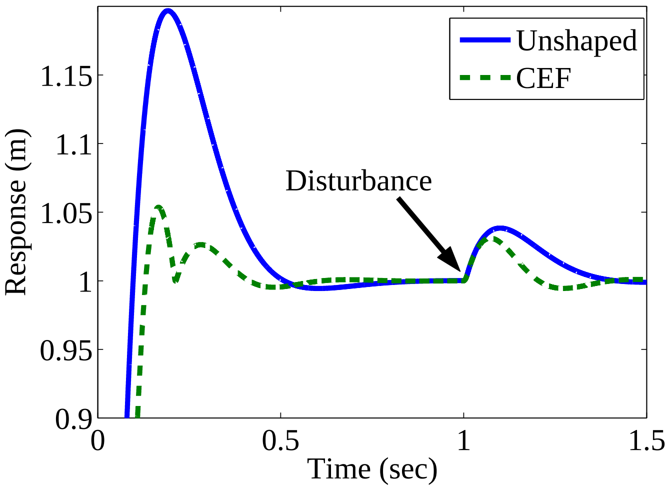


Figure 1.18: PD Control vs. Input Shaping Combined with PD Control.

feedback control, Dharne and Jayasuriya [9], Chang and Park [6], Agostini, *et al.* [1], Magee, Cannon and Book [49], and Seth, Rattan and Brandstetter [63] did not investigate the concurrent design of OLIS and feedback control. The closed-loop system was designed first, and then the input shaper was designed for the already determined closed-loop dynamics. This dissertation seeks a concurrent design scheme with the intent of producing superior controller combinations than is likely with sequential design. It seeks to fundamentally advance the ideas previously advocated by first revealing the underlying reasons why concurrently designing outside-the-loop input shaping and PID control is superior to sequential design. This understanding will then lead to a general framework for concurrently designing input shaping and PID control.

There are, of course, a few cases where concurrent design of input shaping and feedback controllers has been investigated. However, various aspects of these investigations have motivated the research described in this dissertation. For example, several concurrent design schemes do not use PID feedback control, opting instead for more complicated control schemes such as full state-feedback control or Time Delay Control. Time Delay Control is a control strategy that intentionally places time delays within the feedback loop. While these control schemes are widely known, this dissertation seeks to utilize a PID control law because it is simple and practical for a wide variety of applications, and it is the most widely used controller. Some of the current research is also explicitly focused on one particular system or plant type. While this research will start with an in-depth investigation of simple plant types, it will then progress to the concurrent design of OLIS and PID feedback control for a variety of more complicated plant types including multi-mode systems and systems with numerator dynamics.

Many of the current design schemes also utilize a complicated nonlinear optimization technique. Some of these optimization techniques leave the input shaper parameters completely open; that is, nothing is known or assumed about the impulse times or amplitudes. Often, only the number of impulses is chosen before optimization. Other optimization schemes do not specifically ensure that any performance measures (overshoot, actuator limits, etc.) are met. Instead, a cost function routine is used to choose controller parameters. And finally, very few of the concurrent design schemes reviewed here investigate why and how the concurrent design of OLIS and PID feedback control yields different and superior control schemes. Two exceptions to this are [41] and [34] who briefly mention that concurrently designing OLIS/PID controllers would allow for higher PID gains and, consequently, faster responses.

This thesis develops an advancement in the methodology for concurrently designing input shaper and PID parameters. First, the underlying principles that make concurrent design better than the traditional sequential design will be revealed. This knowledge is expected to enable more intelligent design of future solution methods. For example, this thesis will use this knowledge to develop a new input shaper that would not have been an otherwise intuitive choice. Furthermore, this research will mainly rely on standard input shaper forms (like the ZV, ZVD, UMZV, EI and SNA shapers mentioned before and often found in the literature). This will simplify solution routines by eliminating variables while maintaining the primary purpose of input shapers - to reduce vibration. However, it should be noted that narrowing the shaper parameter choices will limit the solution space and possibly result in non-optimal solutions. However, it is believed that these input shapers will result in near-optimal solutions while also greatly simplifying the design routines. Finally, this research will focus on solution routines that specifically set and meet performance constraints so as to yield more practical solutions that are applicable to many real-world scenarios.

1.7.1.2 Research Hypothesis

In the area of concurrently designing OLIS and PID controllers, a general framework useful for a wide range of linear, finite order, causal systems can be developed such that concurrently designed combinations are uniquely different from, and superior to, sequentially designed combinations.

1.7.2 Inserting Input Shapers within Feedback Loops

Most of the research addressing the use of input shaping filters within feedback loops can be grouped into four distinct categories.

Classical Method

Model Reference Controller

Plant Inversion Method

Quasi-CLSS Controllers

The first type of CLSS controller is called the "Classical Method" and refers to the block diagram shown in Figure 1.19. In this figure, C is some feedback controller, IS is the input shaping filter, and G is the plant. This form of CLSS is the most intuitive approach (simply place the filter in the feedforward loop) and is the form most often found in the literature. The second form of CLSS is called "Model Reference Controller" and is shown in Figure 1.20. Here, G_a is the actual plant and G_m is the

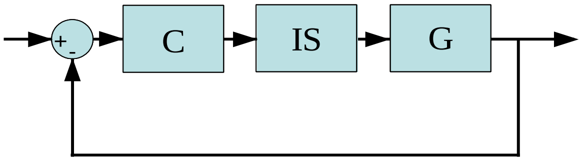


Figure 1.19: Basic CLSS Block Diagram.

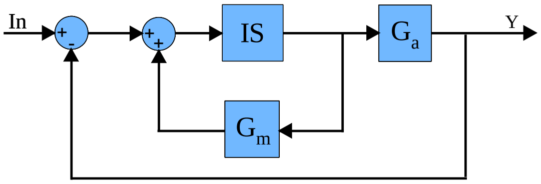


Figure 1.20: Model Reference CLSS.

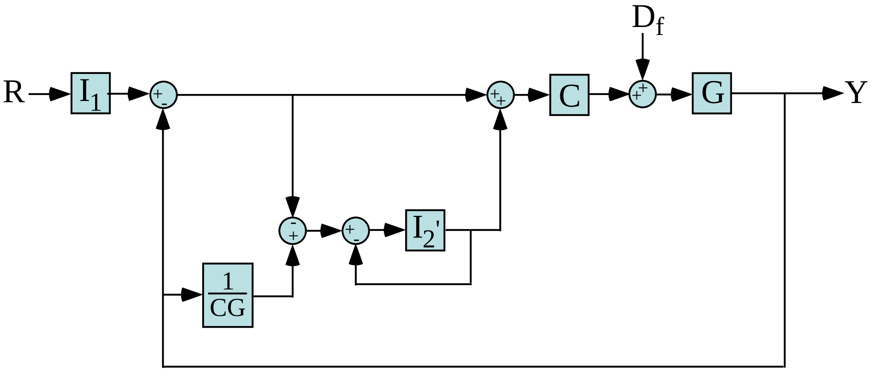


Figure 1.21: Plant Inversion CLSS.

modeled plant. This method relies upon comparing a modeled system output with the actual error signal. This difference is then filtered by the input shaper. The third form of CLSS is called the "Plant Inversion Method" and is shown in Figure 1.21. This method uses an inverted plant/controller model $\left(\frac{1}{CG}\right)$ to determine disturbance inputs so that they can then be filtered by an input shaper (I_2'). Finally, the fourth category is called "Quasi-CLSS Controllers". These controllers use command generation filters within feedback loops but differ fundamentally with CLSS in some respect. For instance, many of these controllers do not send real time feedback signals through input shaping filters, but they do use sensor information to periodically change the shaper parameters. Others do filter real time feedback signals, but the filter is not an input shaper. It is some other type of command shaping filter.

1.7.2.1 Classical Method

Several researchers have studied CLSS controllers which fall into the "Classical Method" category [5, 10, 28, 29, 47, 48, 94, 99, 105, 106]. For instance, Kapila, *et al.* designed a CLSS controller to perform well despite modelling errors and errors in the timing of the shaper impulses [28, 29]. Zuo, *et al.* and Drapeau, *et al.* also designed a CLSS controller [10, 105, 106]. They experimentally compared it to PID control combined with OLIS, as well as experimentally demonstrated their CLSS controller's ability to reject sensor disturbances. Tzes combined a robust LMI (Linear Matrix Inequality) controller with an input shaping filter in the feedforward loop [99]. Finally, Magee and Book implemented a Classical CLSS controller on a robotic manipulator and demonstrated its vibration reduction capabilities, as well as its potential closed-loop stability problems [47, 48].

1.7.2.2 Model Reference Controller

This form of CLSS has received only scant attention [91]. It is nonetheless an interesting idea, because, as mentioned by the authors, the controller is specifically designed to reject force disturbances. However, the authors only briefly mention this application and focus on the stability of the proposed controller in the presence of natural frequency modeling errors.

1.7.2.3 Plant Inversion Method

O.J.M. Smith envisioned a unique CLSS controller that utilizes plant inversion to identify disturbance signals so that they can then be filtered by an input shaper [82–84]. However, upon further investigation, there were several obstacles that needed to be addressed before this control scheme could be practically implemented. The main issue concerned the causality of the inverted plant model. This issue is often dealt with via time delays in the digital domain. However, since the inverted plant model is inside a feedback loop, adding time delays presents a serious stability concern.

1.7.2.4 Quasi-CLSS Controllers

Somewhat bridging the gap between OLIS control and CLSS, Quasi-CLSS Controllers offer a compromise between the two. For example, Chang, *et al.*, devised a logic controller that monitors the system output for disturbances [8, 59]. When a disturbance is detected, a reaction force is given by the actuator to cancel the vibratory effects of the original disturbance. Park and Chang also developed an adaptive input shaping controller for non-LTI systems [60, 61]. This controller repetitively sends reference commands, measures the output vibration, and then adjusts the shaper parameters so that the next reference signal will result in less vibration. Pao and La-orpacharapan developed a logic controller that uses feedback information to determine switch times for an input shaped signal [56]. Finally, Tzes and Yurkovich used a frequency domain identification scheme to calculate the input shaper time locations [100].

The second type of Quasi-CLSS Controller does continuously filter a feedback signal, but the filter is not an input shaper. Zhong and Hang developed such a controller in the digital domain for a first order plant with a time delay [104].

1.7.2.5 Closed-Loop Stability

The literature presents a few basic guidelines for achieving stability in CLSS controllers. For instance, when analyzing a special class of manipulators, Zuo et al. established, via a Nyquist analysis, a desired relationship between the system's crossover frequency (ω_c) and frequency of vibration (ω_n) to ensure closed-loop stability [105, 106]. Calvert and Sze [5, 94] and Smith [84] also developed a Nyquist criterion based approach to determine stability. Kapila, *et al.*, used Lyapunov stability criterion to show that their particular CLSS strategy will be asymptotically stable [28, 29]. Finally, Staehlin and Singh, who analyzed an undamped, second-order system, discussed closed-loop stability in terms of the relationship between the modeled and actual system frequencies [91].

1.7.2.6 Applications of CLSS

Many authors speculate that CLSS controllers will enhance disturbance rejection, effects of modeling errors and nonlinearities. However, only a few have numerically or experimentally verified these assumptions. For example, Kapila, *et al.*, compared outside-the-loop input shaping and closed-loop signal shaping when errors in the timing of shaper impulses was considered [28, 29]. In addition, Zuo, *et al.*, showed that CLSS can reject sensor disturbances and provide improved trajectory tracking when compared to traditional PID control [105].

1.7.2.7 Research Gap

Considering input shaping's success in working outside the feedback loop, it is natural to think that it has potential to improve a system's response to disturbances, non-zero initial conditions, etc. by including it within a feedback loop. However, input shapers partially delay the signals that pass through them. Considering the basic knowledge of how full time delays affect closed-loop stability, the use of partial delays certainly presents a stability question. Unfortunately, the literature lacks an in-depth presentation and understanding of the stability of feedback systems utilizing input shapers inside the loop. For instance, many current stability investigations are specific to one type of plant model. What is needed is a basic and intuitive understanding of the stability characteristics inherent to CLSS controllers.

The literature also lacks a detailed investigation into the potential uses of CLSS controllers. While some researchers have revealed several advantages of CLSS, there is still much work to be done in this area. One major area of investigation is force disturbance rejection. Both Staehlin [91] and Smith [84] proposed unique CLSS controllers for force disturbance rejection. However, these controllers are somewhat impractical or ineffective in their current forms. There are also other areas that need to be investigated: such as non-zero initial conditions, nonlinearities and modeling errors.

1.7.2.8 Research Hypothesis

1) Because input shaping only partially delays an incoming signal, a feedback controller containing an input shaping filter can be designed to be stable within a range of parameter uncertainty.

2) Utilizing an input shaper within the feedback loop will be useful and advantageous so long as it fully filters any signal which it is intended to act upon.

1.7.3 Literature Review - Similar Strategies

The area of controls has been extensively studied for many years. Therefore, new control theories need to be discussed within the context of, and differentiated from, similar, established methodologies. In particular, the area of CLSS is being presented as a recently new control strategy. In this section, CLSS will be compared to, and contrasted from, "Loop Shaping", "Zero Phase Error Tracking Control" and "Time Delay Control".

1.7.3.1 Loop Shaping

The basic premise of loop shaping is to take some plant and combine it with a controller such that the open-loop frequency response fits within specified boundaries set by the closed-loop system's desired performance characteristics and limitations. In some sense, any kind of feedback control can be viewed in this light. Even the area of closed-loop signal shaping can be viewed as the desire to reduced an oscillatory system's frequency spike via the addition of a corresponding frequency trough.

Loop shaping is a well known feedback controller design strategy that has received a significant amount of attention. Del Vescovo and D'Ambrogio used a double feedback loop and shaped both loops [102]. They used a rational transfer function but claimed that the controller can be of any form. Eberhardt and Saridereli developed their own loop shaping technique that has the advantage of utilizing a simple computation algorithm [11]. Sparks, Banda, and Yeh compared several loop shaping techniques which include the use of LQR, H_∞, and H_2 optimization routines [90]. Grassi and Tsakalis used an H_∞ optimization technique to choose PID gains for a SISO system [18]. They also discussed the use and optimization of rational pre-filtering transfer functions similar to the optimal combination of OLIS and PID feedback control. Grassi, *et al.*, then extended their technique to make it adaptive and self-tuning [19]. Finally, Smith and Messner used modified frequency plot techniques called "fsbode" and "ftbode" to optimize controller gains for a digital FIR notch filter [86].

There are several major differences between Loop Shaping and CLSS. First, Loop Shaping generally seeks to alter a large portion of an open-loop system's frequency response - sometimes in both magnitude and phase. CLSS only seeks to alter part one, small part of an open-loop system's magnitude plot. It seeks to diminish (or eliminate) the peaks associated with oscillatory dynamics. CLSS is generally unconcerned about specifically altering the phase of the open-loop system or any of the remaining portions of the magnitude response. Second, Loop Shaping is primarily accomplish through complicated optimization routines. A CLSS filter, on the other hand, will, in general, be easier to design because (similar to standard input shaping) it is primarily based upon the original open-loop system's flexible modes. Any secondary controllers can be designed via standard root locus or Bode design tools. Thirdly, CLSS filters are inherently irrational transfer functions, consisting primarily of time delays. Most Loop Shaping controllers are designed from rational transfer functions (often PID controllers). Lastly, because of its complexity, Loop Shaping is often reserved for complicated applications like MIMO systems. CLSS is designed for SISO systems, although it would be easily applicable to uncoupled, MIMO systems.

1.7.3.2 Plant Inversion, Zero Phase Error Tracking Control (ZPETC)

The idea of plant inversion is a very attractive concept within the controls field. It promises the ability to calculate the forces needed to exactly achieve any desired motion. However, this method has several issues which make its use on real-world systems problematic. One, it requires an exceedingly accurate model of the system being controlled. Second, aggressive motions can result in actuator demands that exceed the capabilities of most real-world systems. Third, in its most basic form, it is incapable of successfully controlling non-minimum phase systems – which occur quite often in the real-world. Fourth, it often requires some advanced knowledge of the desired trajectory, which is not always possible (or, at least, is undesirable) in practical applications.

There has been some work on advancing the idea of plant inversion to address some of these shortcomings. Kao, Sinha, and Mahalanabis use a nonlinear feedback controller on a nonlinear plant to force the closed-loop poles to some desired location [26]. An inversion of the closed-loop system is utilized as a pre-filter. To address the problem of non-minimum phase plants, Tomizuka developed Zero Phase Error Tracking Control (ZPETC) [95]. While this control scheme works for non-minimum phase plants, it does require some future input knowledge and is sensitive to modeling errors. To address the issue of modeling errors and time-varying systems, Tsao and Tomizuka developed an adaptive ZPET controller [96]. While effective, there are always some implementation difficulties associated with adaptive feedback control. Finally, Smith and Lee experimentally compared non-adaptive and adaptive ZPET controllers [80]. The adaptive version outperformed the non-adaptive controller in the experimental results. However, the inputs were again complete trajectories known before the start of any motion. In addition, both control schemes required repetitive tuning. The adaptive controller even had some computation problems with the parameter estimation. Finally, Wen and Potsaid compared ZPETC to other feedforward controllers for use in an adaptive control scheme [103].

Zero Phase Error Tracking Control is uniquely different from Closed-Loop Signal Shaping. ZPETC is an outside-the-loop controller which seeks to improve a system's ability to follow a desired reference command. Sometimes, in the adaptive versions, feedback is used to adjust the ZPETC gains in real time. On the other hand, CLSS specifically seeks to place input shapers within feedback loops in order to address a wider variety of controls problems. These problems include disturbance rejection, system nonlinearities and non-collocated control.

ZPETC is similar to outside-the-loop input shaping, in that both of these control techniques seeks to improve a system's response to a reference command. ZPETC is a more complicated control technique, often requiring an adaptive version in practical applications. However, it is well-suited for applications where complex trajectories must be accurately followed. OLIS is a much simpler control technique that is easy to implement. However, it has not been extensively used to aid a system in following a complex trajectory. Its typical usage is with simple reference commands, like step inputs. Because this dissertation will focus on simple commands (like step inputs), input shaping (both inside and outside the loop) will be studied.

1.7.3.3 Time Delay Control (TDC)

The research addressing closed-loop systems with time delays seems to fit into two main categories. The first category looks at closed-loop systems with inherent time delays, often in the feedback loop. The goal here is to add some additional controller to these systems so that various stability and performance constraints are met. The second category seeks to intentionally add time delays within a closed-loop architecture (often when no time delays were originally present) so as to again guarantee some level of stability and performance measures. Most of the reported research focuses on ensuring stability.

The majority of controller design schemes found in the literature are highly mathematical and complicated, relying on Ricatti equation and Lyapunov stability based design. Lee, *et al.*, implemented a hybrid impedance/time-delay controller on a robotic manipulator [42]. Niculescu, Fu, and Li developed a Lyapunov-Razumikhin and a frequency-filtering state-space controller for systems with time delays [53]. Haddad, *et al.*, Mahmoud and Al-Muthairi, and Kapila, *et al.* developed Lyapunov and/or Ricatti based controllers [21, 27, 50]. Mahmoud and Bingulac developed a robust stabilizing scheme for interconnected, uncertain systems [51].

Some other research takes Time Delay Control a step further, intentionally adding delays to achieve closed-loop stability [2,25,35,101,104]. The most interesting of these are the works done by Kumar, Alli, Kang and Udwadia who investigated the use of time delays to stabilize non-collocated systems [2, 25, 35, 101].

Finally, a few researchers have gone beyond stability to include performance specifications in their controllers: including Haddad, *et al.*, and Alli and Singh [2, 21].

There are three main differences between Time Delay Control and Closed-Loop Signal Shaping. First, TDC generally uses full time delays when intentionally adding delays as part of a control law. CLSS generally uses partial time delays, allowing some portion of the error signal to pass through the controller un-delayed. One exception to this is a modified version of the Model Reference CLSS that will be presented in Chapter 8. The second main difference is that Closed-Loop Signal Shaping specifically uses partial time delays to form a filter for the sole purpose of eliminating an oscillatory vibration mode. Time Delay Control controllers are primarily concerned with achieving closed-loop stability. Thirdly, the methods used in TDC to guarantee stability are generally quite complex (usually full Lyapunov stability proofs). This research into the field of CLSS has intentionally utilized simple and intuitive stability analysis tools like the root locus and Bode diagram.

CHAPTER II

CONCURRENT DESIGN OF OUTSIDE-THE-LOOP INPUT SHAPING AND PD FEEDBACK CONTROLLERS - MASS PLANTS

As mentioned in Chapter 1, the concurrent design of outside-the-loop input shaping and Proportional-Integral-Derivative feedback control in this thesis will concentrate on the control structure depicted in Figure 2.1. The concurrent design strategy will calculate the PID controller gains and the input shaper ("IS") parameters simultaneously. This strategy will be compared to the state-of-the-art sequential design scheme, which first determines the PID controller gains independent of the effects that the input shaper will produce. Only then is an input shaper added to the sequentially designed closed-loop system to further decrease settling time and vibration.

This chapter will focus on the control of a mass. Also, the feedback controller used will be restricted to a PD type controller. The integral action is avoided here for two reasons. One, a mass under PD control naturally results in zero steady-state error when given a step reference command. Also, the addition of the integral action will

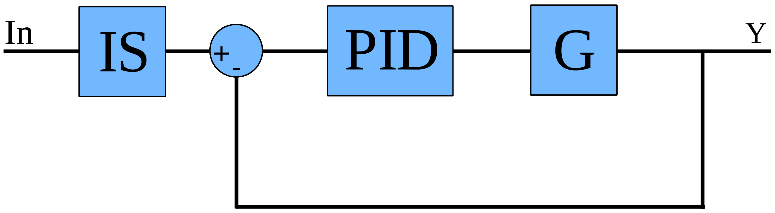


Figure 2.1: Basic Block Diagram for Concurrent Design.

add a first-order component to the closed-loop transfer function. The simple design strategy taken in this chapter is based upon having a pure second-order closed-loop transfer function. Integral control will be added to the feedback controller and design strategy in a later chapter.

2.1 Basic Assumptions

The following four chapters will discuss the concurrent design of outside-the-loop input shaping (OLIS) and PID feedback control. For the research presented here, the following basic assumptions are made:

1. The plant (G) is stabilizable via PID feedback

2. G is linear, time-invariant

3. G is minimum phase

4. The closed-loop system is a single-input, single-output (SISO) system

Each particular study within the following four chapters will add additional assumptions that are detailed within their respective section. While assumption #1 will always be required, future work could extend this research beyond the constraints set by assumptions #2 - #4.

2.2 Design Methodology

There are many basic methods by which a control system can be designed and controller parameters chosen. For relatively simple systems, analytical design tools can be used to find a solution. For more complicated systems, various optimization techniques or search routines can be used. Each of these solution methods has its own advantages and disadvantages.

The research discussed in this dissertation is primarily concerned with proving the superiority of concurrent design of OLIS and PID feedback control over a sequential design methodology. Therefore, the particular design strategy used (analytical, nonlinear optimization, search routine, etc.) is somewhat irrelevant. The methodology chosen for this research uses a basic knowledge of input shaping, feedback control, and linear system behavior to create relatively simple concurrent design methodologies that yield practical solutions for real-world applications. The remainder of this section will use several design schemes from the literature to describe the details of the design methodology chosen for this thesis, as well as to highlight the motivation for such a choice.

2.2.1 Rigid vs. Soft Constraints

In 2005, Gopalakrishnan, Reddy, and Singh published a paper describing their methodology for concurrently designing OLIS and PD feedback control for second-order systems [16]. Their procedure centered on choosing P and D gains which minimized the cost function

$$J = \int_0^T (1 + \alpha u^2) dt \tag{2.1}$$

when a step reference input was given to the system. Here, T is the time of the final input shaper impulse, α is a weighting constant and u is the actuator effort.

The idea of designing control schemes based on cost function minimization is well established and has been used extensively and successfully. However, as with any design procedure, it does have some weaknesses. Primarily, the fact that J itself has no real, physical meaning can present some problems. For example, it can be shown from (2.1) that one purpose of minimizing J is to minimize the settling time of the controller, whereas another goal is to minimize the actuator effort required. Note that because modeling errors were not investigated in [16], the time of the last shaper impulse, T, is a conservative measure of settling time. However, this procedure is incapable of meeting any strict, numerical limitations on settling time and $u(t)$.

Three examples will be presented here to demonstrate this deficiency. Each example will adopt a different method for finding the shaper and controller parameters and will note the effect on the cost function, J.

The following examples all involve the use of Specified-Negativity-Amplitude (SNA) shapers [69]. As described in Chapter 1, SNA shapers are similar in form to $UMZV$ shapers. Both shapers have three impulses in the following pattern: positive impulse - negative impulse - positive impulse. However, for SNA shapers, the amplitudes of each impulse can vary. Normally, one impulse amplitude is chosen and the remaining two are solved for by using vibration constraint equations. The equations shown in Chapter 1 first specified the amplitude of the negative impulse. Here, however, the amplitude of the first, positive, impulse is varied between 1 and 0.5. This actually creates a spectrum of input shapers ranging from the $UMZV$ (when the first impulse amplitude is unity) to the ZV (when the first impulse amplitude is 0.5). The three examples presented here seek to investigate the usage of these SNA shapers within the cost function minimization design scheme presented by Gopalakrishnan, *et al.* Note that only PD control of a unit mass is investigated.

The first example seeks to minimize the cost function shown in (2.1). Following the intuitive results presented by Gopalakrishnan, *et al.*, the derivative gain is held at zero. Their paper did not analyze the effect of disturbances, so the results discount the need for damping when external disturbances occur. Then, the proportional gain, K_p, and the amplitude of the first shaper impulse, A_1, are varied to see their effect on J. Figure 2.2 shows the natural log of J as a function of both K_p and A_1. Figure 2.3 shows the minimum J value for each A_1 value, and Figure 2.4 shows the settling time of each system response when the optimal A_1 and K_p combinations are used. Figure 2.5 shows the corresponding maximum actuator effort ($max[u(t)] = U_{max}$).

The main result here is that even though the value of J increases along with A_1, as shown in Figure 2.3, the actual U_{max} reached during the step response generally

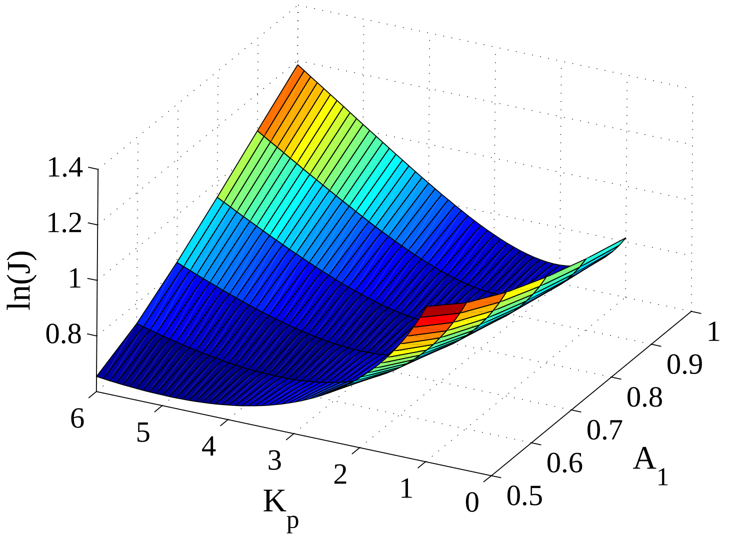


Figure 2.2: J as a Function of K_p and A_1.

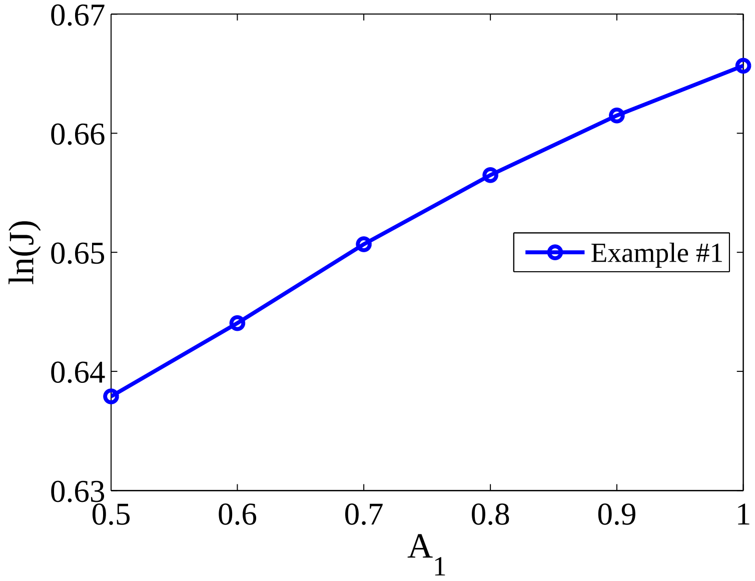


Figure 2.3: Minimum J Value as a Function of A_1.

decreases with an increase in A_1. Also, the increasing J does not necessarily mean a decrease in the speed of the system, as shown in Figure 2.4.

As a second example demonstrating the major problem with cost function minimization, the PD gains and input shaper parameters were not chosen so as to minimize

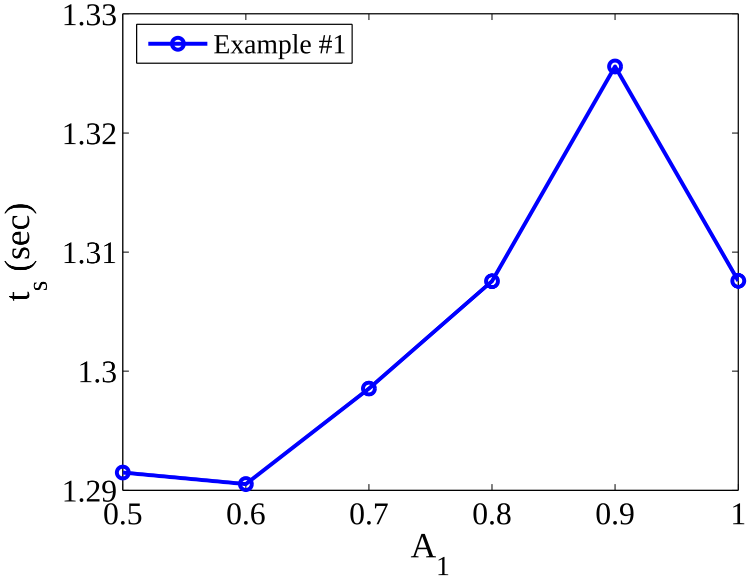


Figure 2.4: Settling Time of K_p & A_1 Optimal Combinations.

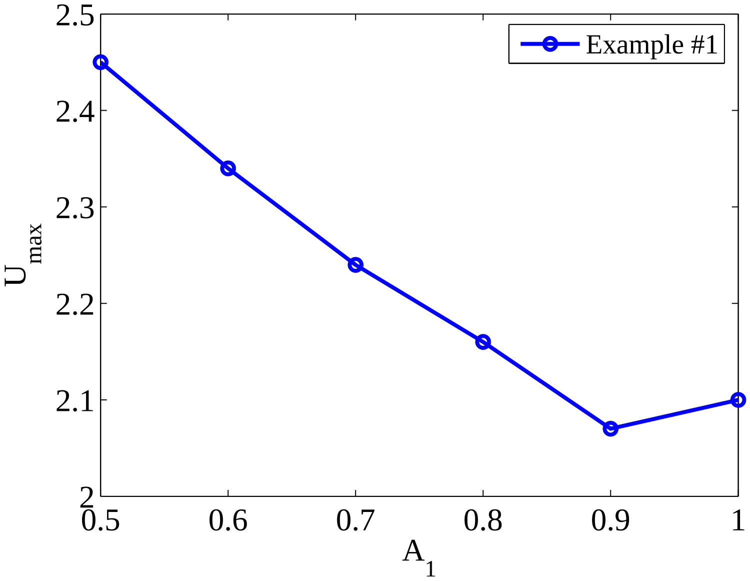


Figure 2.5: Maximum Actuator Requirement for K_p & A_1 Optimal Combinations.

J. Instead, one fixed set of K_d (zero) and K_p values established in [16] is used on the same range of SNA shapers bridging the gap between the ZV and $UMZV$ shapers. Again, the effects on J, settling time, and actuator effort are noted. As shown in Figure 2.6, as A_1 increases, J also increases. This is the same trend found in Example #1.

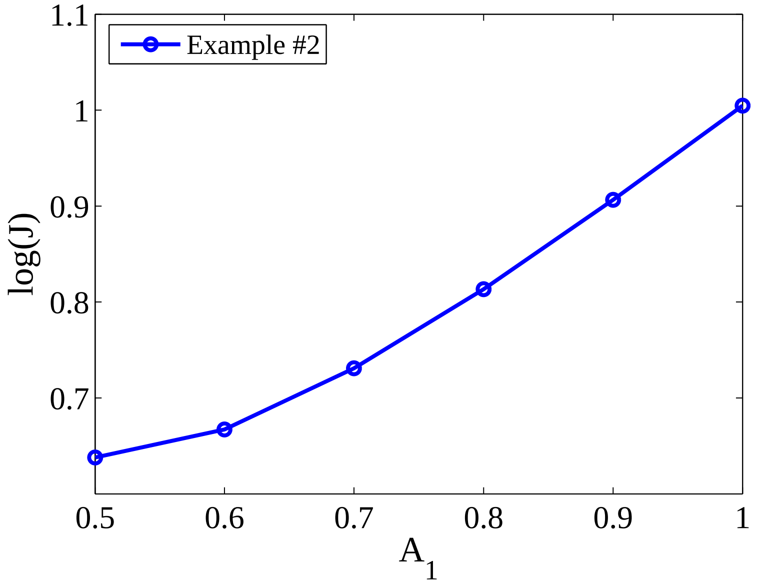


Figure 2.6: J as a Function of A_1 for the Original K_p Value.

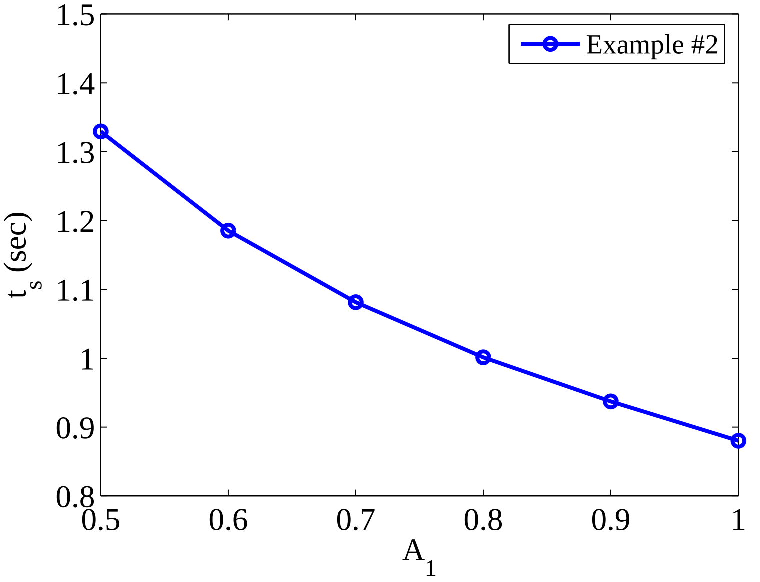


Figure 2.7: SNA Shaper Settling Times Under the Original K_p Value.

However, the effects on settling time and U_{max} are quite different. As shown in Figure 2.7, as the SNA shaper approaches the $UMZV$ shaper (i.e. A_1 increases from 0.5 to 1), settling time uniformly decreases. The actuator effort required for this speed increase is shown in Figure 2.8. Here, it is clear that the more aggressive shapers

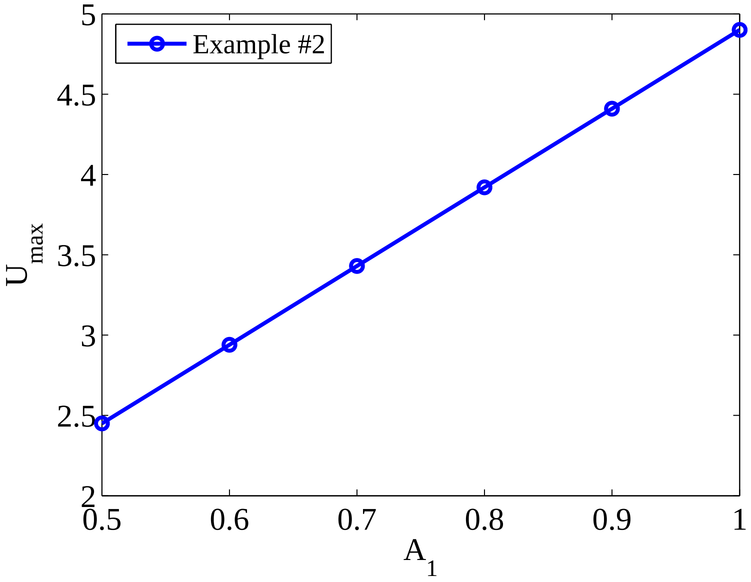


Figure 2.8: *SNA* Shaper Maximum Actuator Efforts with Original K_p Value.

require a higher U_{max}. To summarize the results of this example; J increased, rise time decreased and U_{max} increased with an increase in the amplitude of the shaper's first impulse, A_1.

Finally, a third example is presented. Again, a range of SNA shapers is investigated which spans the gap between ZV and $UMZV$ shapers. However, unlike the second example, K_p is not held constant. And, unlike the first example, K_p is not chosen so as to minimize J. Instead, for each A_1 value chosen, K_p is uniquely determined such that U_{max} remains constant.

The main result here is that despite the increase in the cost function J, as seen in Figure 2.9, increasing A_1 actually decreases the system rise time, as seen in Figure 2.10, while maintaining a constant U_{max}. This means that the system responds faster despite no increase in the maximum required actuator effort. In terms of these real, physical characteristics, this is a definite improvement with little cost.

Of course, the seemingly conflicting results of the three previously described investigations can be explained by noting again that the cost function J has no direct, physical meaning. Because it has no physical meaning and is actually a mathematical

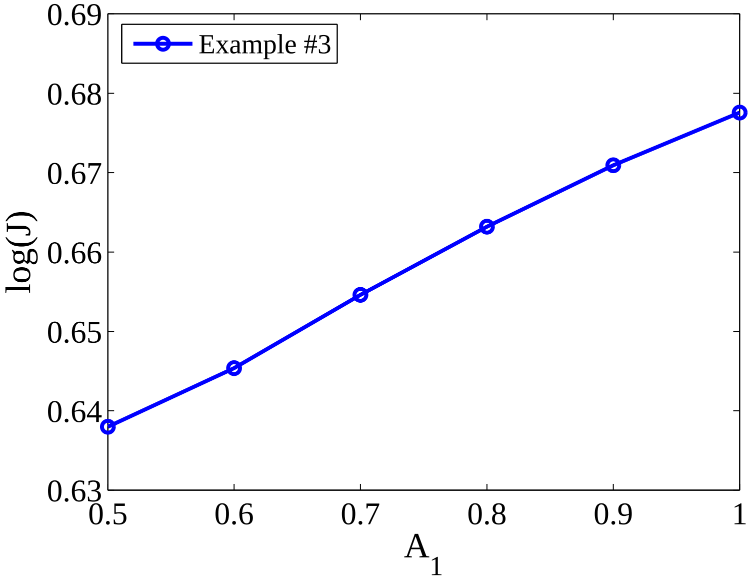


Figure 2.9: J as a Function of A_1 for the New K_p Values.

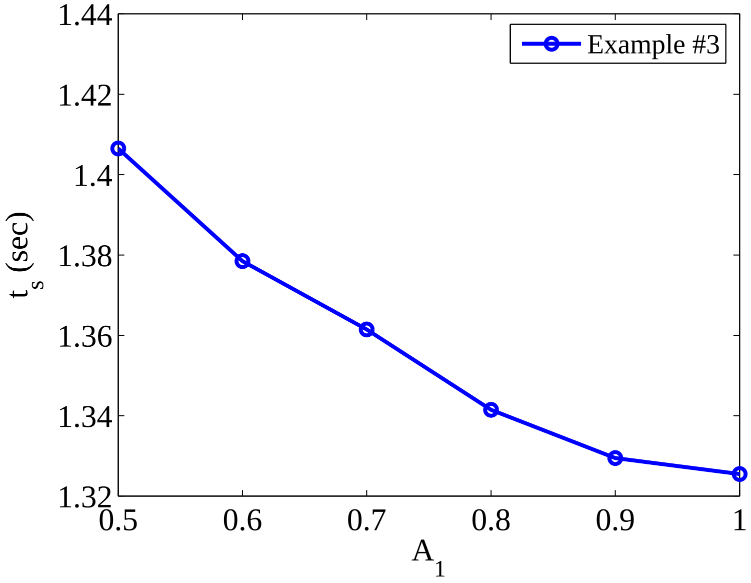


Figure 2.10: SNA Shaper Settling Times Under New K_p Values.

combination of the settling time and actuator effort, simply noting a numeric trend in J is not sufficient to say anything definitive about the effect on settling time or actuator effort. This is verified by the summary of findings in Table 2.1. For this reason, relying upon this type of cost function design procedure can be problematic when the actual, numeric values of a system's physical characteristics are important.

Table 2.1: Summary of Results for the Three Cost Function Examples

A_1	J	t_s	U_{max}
Increased	Increased	Generally Increased	Generally Decreased
Increased	Increased	Decreased	Increased
Increased	Increased	Decreased	Equal

It is important to note that the cost function optimization presented by Gopalakrishnan, *et. al*, does have many practical uses. For example, the cost function they introduce does have a direct measure of the energy utilized throughout the system's motion, even if it cannot specifically limit the maximum actuator effort reached during the move. This is clearly an important issue in many situations. However, this dissertation will focus on applications where the primary objective is to move as fast as possible while maintaining meaningful performance constraints on such things as overshoot and maximum actuator effort. Therefore, this research will avoid the use of cost function minimization where the cost function itself lacks physical meaning.

2.2.2 Derivative Control vs. Velocity Feedback

Figure 2.11 shows a typical form of PD control where both the proportional and derivative control elements lie in the forward loop. However, this form of PD control presents two major challenges. One, most of the research (including this research) on concurrently designing OLIS and PID controllers has focused on simple, step commands. When step commands, or any other highly aggressive commands, are given to a control scheme like the one depicted in Figure 2.11, the derivative portion of the controller will contain large spikes due to differentiating such aggressive commands. This can lead to unrealistic actuator effort requirements. Secondly, the use of derivative control in the forward loop results in a closed-loop zero deriving purely from the controller, as is seen in the following closed-loop transfer function:

$$tf_1 = \frac{(K_p + K_d s)G}{1 + (K_p + K_d s)G} \tag{2.2}$$

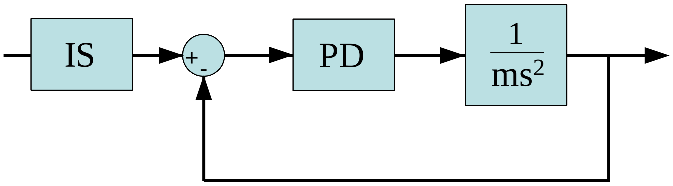


Figure 2.11: Standard OLIS/PD Control of a Mass.

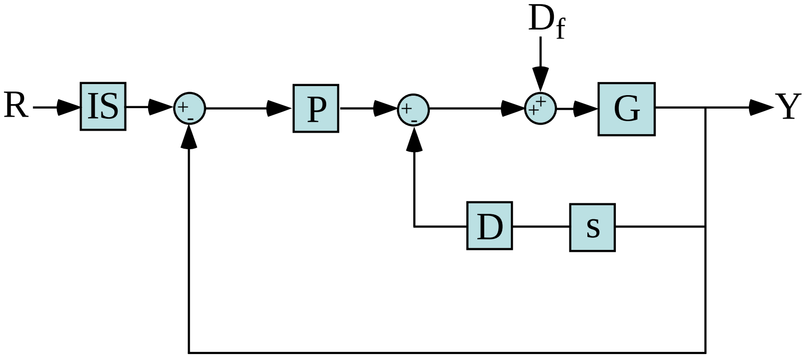


Figure 2.12: Proportional + Velocity Feedback Block Diagram.

For simple systems, like masses and second-order oscillators, the concurrent design procedure can largely be done analytically. However, the presence of closed-loop numerators can complicate this process or even make an analytical solution unachievable, because the well known equations that describe the motion of simple, second-order systems begin to break down when numerator dynamics are present.

The concurrent design scheme proposed in this dissertation will utilize velocity feedback as its manifestation of derivative control. Shown in Figure 2.12, velocity feedback puts the derivative control action in the feedback loop and never attempts to differentiate the reference signal R. This avoids both of the problems listed above. Aggressive reference signals do not result in large actuator effort spikes, and this form of the PD controller does not add any closed-loop zeros. This is seen in the following closed-loop transfer function:

$$\frac{Y}{R} = \frac{K_p G}{1 + (K_p + K_d s)G} \tag{2.3}$$

Note that Gopalakrishnan, *et. al*, successfully utilize this type of PD controller to simplify their concurrent design scheme [16].

2.2.3 Input Shaper Limitations

Another important issue concerns the choice of input shapers used in this research. Kenison and Singhose utilized a three-impulse shaper that was virtually unconstrained [30, 31]. While the time of the first impulse was held at zero, the amplitudes of all three impulses and the timing of the final two impulses were variable. Unfortunately, with the addition of the closed-loop controller's proportional and derivative gains, this meant that their optimization routine had to determine seven variables. This resulted in an optimization routine with many local minima.

On the other hand, Gopalakrishnan, *et. al*, looked at input shapers that can all be expressed as multiple convolutions of the basic ZV shaper [16]. That is, they investigated the ZV shaper, the ZVD (two ZV shapers convolved together), the ZVDD (three ZV shapers convolved together), etc. The advantage to doing this is that the input shaper parameters have less freedom, thereby simplifying the solution routine. However, by restricting the choice of input shaper parameters, the solution space is also restricted, meaning that there is the potential for optimal solutions to be missed.

The research described here seeks to form a compromise between these two extremes. Input shapers are restricted to fit within certain, basic formats to allow for near-analytical solutions and quick search routines. For example, the ZV and ZVD shapers are often used. However, this dissertation will also often use input shapers that have one independently varying parameter. This means that there are multiple shaper parameters possible for any given closed-loop system, thereby allowing for a larger solution space to be spanned in search for the optimal, or near-optimal, solution. These input shapers will be described in more detail throughout the following chapters.

2.2.4 Limitations of Concurrent Design

Before describing the design procedures and results arising from this research, it is important to note the three primary reasons why the concurrent design of OLIS and PID controllers will fail to be superior to sequential design of OLIS and PID controllers. These reasons are general to the concurrent vs. sequential design argument and are not unique to the particular design procedures presented in this dissertation.

First, PID control cannot be used on every linear, causal plant. For example, PID control cannot stabilize the plant $G = \frac{1}{s^6}$. In the cases where PID feedback cannot stabilize the plant, both sequential and concurrent design of OLIS and PID feedback control are equally useless.

The second area in which both concurrent and sequential design are equally inept is the case when the design specifications are unachievable. For example, if the constraints state that a system must move extremely fast with little actuator effort, this simply may not be possible. This type of situation will often leave the concurrent and sequential design equally incapable. However, one interesting result in this dissertation will show that there are situations where sequential design will not find a solution, but concurrent design will. This result suggests the superiority of a concurrent design scheme, even if it cannot always find a solution.

Finally, the third situation occurs when sequential and concurrent design yield the same solution. One example of this occurs when the design constraints force the closed-loop system to be non-oscillatory. Here, there is no need for an input shaper. Therefore, the concurrent design scheme becomes equal to the sequential design scheme.

2.3 Design of OLIS and PD Feedback Controllers

To begin the study of concurrently designing outside-the-loop input shaping and PD feedback control, the problem of moving a mass is revisited. This problem was addressed by Kenison and Singhose with a full nonlinear optimization routine [30,31], as well as by Gopalakrishnan, *et. al*, with their cost function minimization routine [16]. Here, a less complicated optimization routine is implemented. Because the plant is a mass, the closed-loop system under PD control is a second-order oscillator. The characteristics of this system's response (overshoot, time constant, settling time, etc.) can be well-defined analytically. This enables a quick, analytically based controller design routine.

The equations governing the system under investigation are:

$$G = \frac{1}{ms^2} \tag{2.4}$$

$$\frac{Y}{R} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.5}$$

$$\omega_n = \sqrt{\frac{K_p}{m}} \qquad \zeta = \frac{K_d}{2\sqrt{mK_p}} \tag{2.6}$$

where G is the transfer function of the plant, m is the mass to be controlled, $\frac{Y}{R}$ is the closed-loop transfer function, ω_n is the closed-loop natural frequency, ζ is the closed-loop damping ratio and K_p and K_d are the proportional and derivative gains.

2.3.1 Sequential Design

In the sequential design process, the closed-loop system is designed without knowledge of the input shaper which will filter the incoming reference command. The response characteristics taken into account in this example include maximum actuator effort, U_{max}, percent overshoot, M_p, settling time with respect to a reference input, t_s, and time constant, t_c. The time constant constraint is used to control the response to an impulse-like disturbance. The equations for these characteristics, considering only the effect of the PD controller, are as follows:

$$U_{max} = mL\omega_n^2 = K_pL \tag{2.7}$$

$$M_p = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{2.8}$$

$$t_s = \frac{4}{\zeta\omega_n} \tag{2.9}$$

$$t_c = \frac{1}{\zeta\omega_n} = \frac{2m}{K_d} \tag{2.10}$$

Here, L is the size of the step reference command. It should also be noted that these equations are only true so long as the closed-loop system remains oscillatory.

For this system, and for a given step move, L, the maximum actuator effort depends only on K_p. Therefore, to maximize the performance of the sequentially designed controller, K_p was set to its maximum possible value. The remaining variable, K_d, was then chosen to achieve the M_p, t_s and t_c constraints.

For the simulation results presented here (sequential and concurrent design), the 2% settling time was prioritized. That is, while the overshoot and time constant constraints were always met, settling time was always minimized within the boundaries formed by the other constraints. Therefore, it can be shown that the optimal sequential design results in a highly damped system ($\zeta = 0.7 - 0.8$). This ζ range for a second-order, closed-loop system is known to result in the quickest settling time in response to a step input. With ω_n set by the pre-chosen K_p value, the chosen ζ value now sets the derivative gain, K_d. According to the above equations, this results in very little overshoot, a relatively high time constant and the lowest possible settling time.

The second step in the sequential design process is to choose the appropriate input shaper to pre-filter the closed-loop system's reference commands. However, according to the first step described above, the closed-loop system has a damping ratio high enough that input shaping provides little to no benefit. So, the second step in the design process would often be to choose no input shaper. However, if K_d was limited in the design constraints, or if rise time t_r was given some priority, the closed-loop system resulting from the sequential design's first step could be oscillatory, with $\zeta < 0.7$. For example, Ogata's "System Dynamics" textbook states that under normal circumstances “it is preferable that the transient response be sufficiently fast as well as reasonably damped. So, in order to get a desirable transient response for a second-order system, the damping ratio ζ may be chosen between 0.4 and 0.8” [55]. For cases when $\zeta < 0.7$, it would then make sense to apply an input shaper in the second design step. Many of the sequential design schemes studied in this dissertation will compromise between rise time and settling time. This will create sequentially designed OLIS/PID control schemes that are more difficult for a concurrent design scheme to outperform.

2.3.2 Concurrent Design

When the use of the input shaper is taken into account, the constraint equations change. As an example, a standard *ZV* shaper will be used here. The equation for U_{max} is now:

$$U_{max} = mA_1L\omega_n^2 = A_1LK_p \tag{2.11}$$

where A_1 is the size of the first input shaper impulse whose equation is:

$$A_1 = \frac{e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}}{1 + e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}} \tag{2.12}$$

The overshoot equation also changes, being set to zero for all cases because zero modeling error is assumed and because no closed-loop zero is introduced by the proportional-plus-velocity-feedback controller.

$$M_p = 0 \tag{2.13}$$

Modeling errors can easily be addressed by using robust input shapers and by estimating the actual M_p by the worst case amount within the expected error range. This will be investigated in the following chapter.

The final equation to change is the settling time. Because the input shaper will eliminate the residual vibration, the system will be “settled” when the input shaper is complete. Therefore, the settling time is set equal to the time of the final input shaper impulse. For a ZV shaper, this results in:

$$t_s = \frac{\pi}{\omega_d} \tag{2.14}$$

Note that this is actually a conservative estimate of settling time. If the input shaper causes the system to come to rest at L by the time of the final impulse, then the system actually crosses the settling time boundary earlier than the final impulse time. For now, the conservative estimate of the final impulse time will be used. In future chapters, this estimate will be slightly revised, and settling time for the concurrent design procedure will be estimated as 90% of the final impulse time.

The one equation which does not change is the time constant. This performance measure is solely concerned with the closed-loop system and is therefore not affected by outside-the-loop filters. Its equation remains:

$$t_c = \frac{1}{\zeta\omega_n} = \frac{2m}{K_d} \tag{2.15}$$

The optimization scheme for the concurrent design strategy is fairly simple, but very different from the sequential design strategy. Settling time is no longer reduced by moving the closed-loop poles away from the imaginary axis, but by moving them away from the real axis (increasing ω_d). Now, the settling time and time constant constraints no longer drive the solution in the same direction. Decreasing one does not necessarily decrease the other, as was the case in the sequential design scheme.

As a result of the competing t_s and t_c constraints and the priority put on t_s, t_c will be set to its minimum value. This will meet, but not exceed, the time constant constraint. As shown in (2.15), setting t_c will also set K_d for a given value of m. With K_d chosen, U_{max} is now a nonlinear function of K_p (assuming m is a known constant). The highest K_p value allowed by the U_{max} constraint will result in the highest possible ω_n and the lowest possible ζ. This, of course, yields the largest

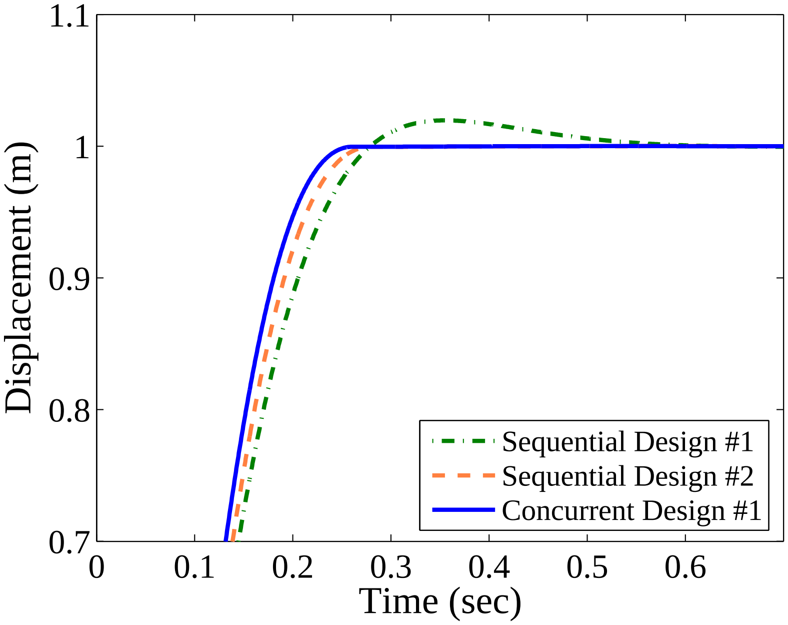


Figure 2.13: Concurrent vs. Sequential Design Step Responses.

possible ω_d and therefore the quickest settling time. Note that because A_1 is always less than unity, the concurrent design scheme will yield higher K_p and ω_n values than the sequential design scheme.

2.3.3 Simulation Results

To verify the superiority of concurrent design, both design schemes were used to create a combination OLIS and PD controller for a mass. The plant was assumed to have unity mass and was required to perform a unit step motion. The actuator effort was required to remain below $200N$ for all time. The maximum overshoot was required to be less than 20%. The minimum time constant, t_c, was 0.116 seconds. Within these boundaries, the settling time, t_s, was minimized.

The step response results of both the sequential and concurrent design strategies can be seen in Figure 2.13. The response labeled "Sequential Design #1" only prioritized settling time. Therefore, the damping ratio produced by the sequential design procedure was $\zeta \approx 0.7$. The system designed according to Section 2.3.2 is labeled as "Concurrent Design #1". As can be seen from this figure, the concurrently designed control scheme is faster than the sequentially designed controller labeled "Sequential Design #1". In fact, the concurrently designed controller has a 2% settling time that is 13% smaller than the first sequentially designed controller.

As was mentioned in Section 2.3.1, another sequential design was performed, and the resulting step response was compared to that of the system created by the concurrent design strategy. The step response to this new, sequentially designed controller is labeled "Sequential Design #2" in each of the figures in this section. This time, the sequential design scheme was required to balance rise time and settling time. This was accomplished by forcing the sequential design to just meet its t_c and M_p constraints, but then remain oscillatory. This kept the closed-loop ω_d relatively high, reducing the system rise time. However, this would technically increase the system's settling time according to the equations in Section 2.3.1. But, since the input shaper will cancel any vibration from this oscillatory system, this new sequentially designed controller actually has a quicker rise time and a quicker settling time than the scheme designed in Section 2.3.1. It should be noted that this presents a somewhat unfair comparison with concurrent design, because this new, sequential design scheme is actually a simplified version of the concurrent design scheme. That is, it took into account the presence of the input shaper and its effect on the overall performance. However, it is used here to reinforce the superiority of fully concurrent design strategies. The step response of the concurrently designed controller is faster than either of the sequentially designed controllers. This modified, sequential design technique will be used several times throughout the following chapters so as to provide a more challenging benchmark by which to measure the performance of a concurrent design technique.

Figure 2.14 shows the disturbance rejection capabilities of the control schemes whose step responses are seen in Figure 2.13. While the first sequentially designed controller does seem to have slightly better disturbance rejection, this is somewhat

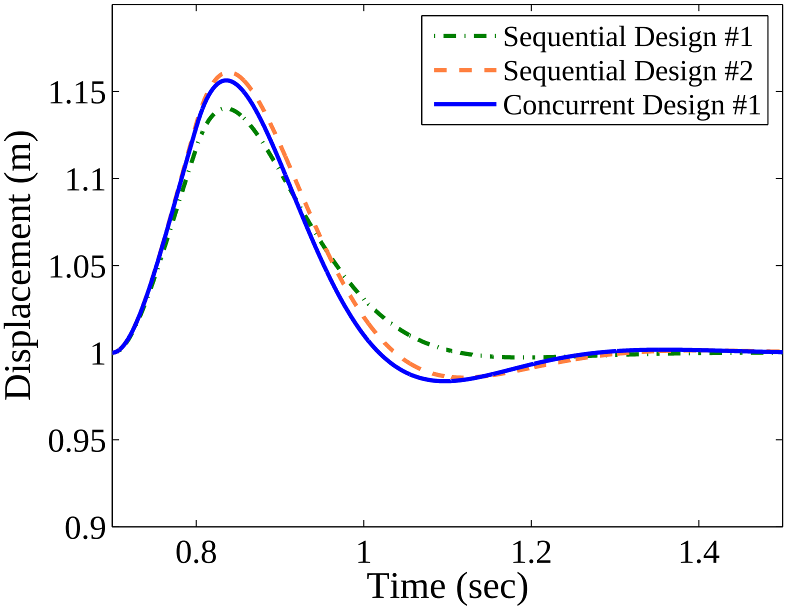


Figure 2.14: Concurrent vs. Sequential Design Disturbance Responses.

irrelevant due to the priority put on minimizing settling time. Each control scheme at least meets the minimum disturbance rejection constraint, t_c. In addition to outperforming the sequentially designed controllers, Figure 2.15 shows that the concurrent scheme was equally able to stay within the actuator saturation limits.

In summary, the two best control schemes ("Concurrent Design #1" and "Sequential Design #2") perform almost identically in response to an impulse-like disturbance. However, the concurrent design strategy still yields a slightly faster response to the step reference input. While both strategies stay within the $200N$ boundary, the concurrent design scheme actually starts closer to $200N$ than the second sequential design does. This occurs because the concurrent design scheme takes into account that the reference input is initially scaled down by the first impulse of the input shaper. This knowledge, combined with the concurrent strategy's full knowledge of input shaping's effect on vibration, allows for higher proportional gains. This results in the faster step response obtained by the concurrently designed controller.

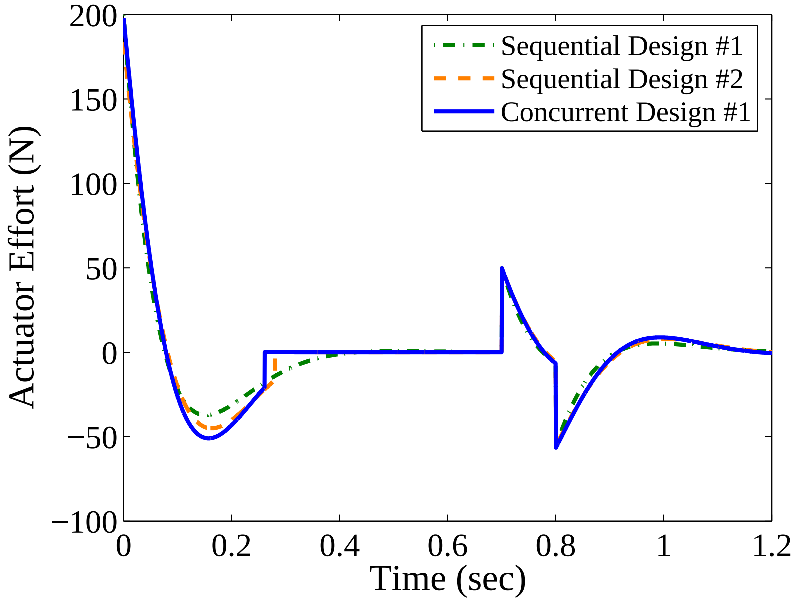


Figure 2.15: Concurrent vs. Sequential Design - Actuator Requirements.

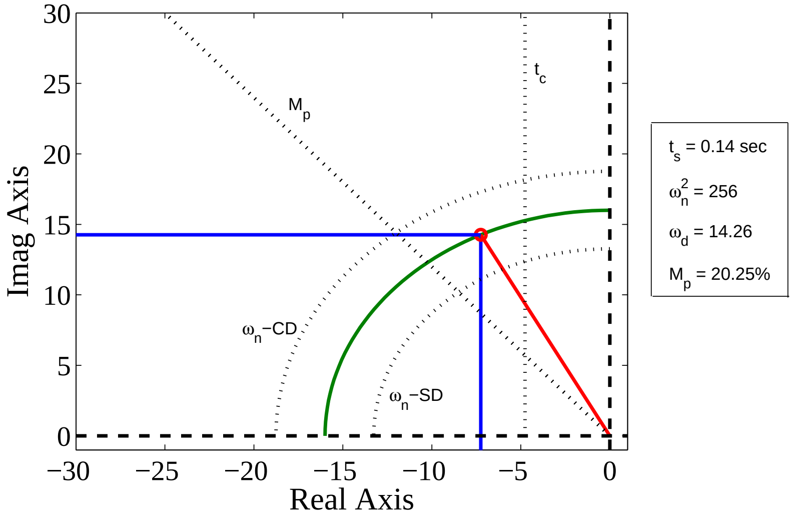


Figure 2.16: GUI Design Tool.

2.3.4 Graphical User Interface

The design schemes presented in this section can be visualized and conducted on a graphical user interface. This is mainly useful for the simplistic mass system, where both design schemes are analytical in nature. Figure 2.16 shows a graphical representation that depicts all of the important system properties and constraints.

The two dashed lines represent the real and imaginary axes. The dotted lines represent the system constraints. The vertical line represents the smallest acceptable time constant. The diagonal line represents the maximum allowable overshoot or minimum damping ratio. The circular lines represent the maximum actuator effort allowed, or the maximum natural frequency. Note that there are two circular lines to depict one of the major differences between concurrent and sequential design. One quarter arc (concurrent design) is larger than the other arc (sequential design) because the concurrent design scheme knows that the initial step is always scaled by the size of the first input shaper impulse, A_1. This means that a concurrent design scheme allows for a larger closed-loop natural frequency than is allowed by a sequential design strategy.

The user interacts with this GUI via the directional keypad buttons and places the closed-loop poles to any desired location. The numerical information on the right then informs the user of important closed-loop system characteristics.

Not only is the GUI useful for replicating the concurrent and sequential design schemes described earlier, it is also useful for adding additional tradeoffs. For example, it is useful for visualizing the sequential design scheme tradeoff between rise time and settling time. In a concurrent design scheme, it can be used to easily study the tradeoff between t_c and t_s.

2.3.5 New Input Shaper with One Independent Variable

While the concurrent design scheme previously described outperformed both of the sequential design schemes, it can be further improved. This is done by abandoning the strict limitation that the shaper adhere to the traditional ZV form. Here, a new input shaper will be derived which is very similar to the ZV shaper, but has one independent variable which is used to increase the solution space and achieve better solutions.

The key to this new input shaper comes from (2.11). The maximum actuator effort

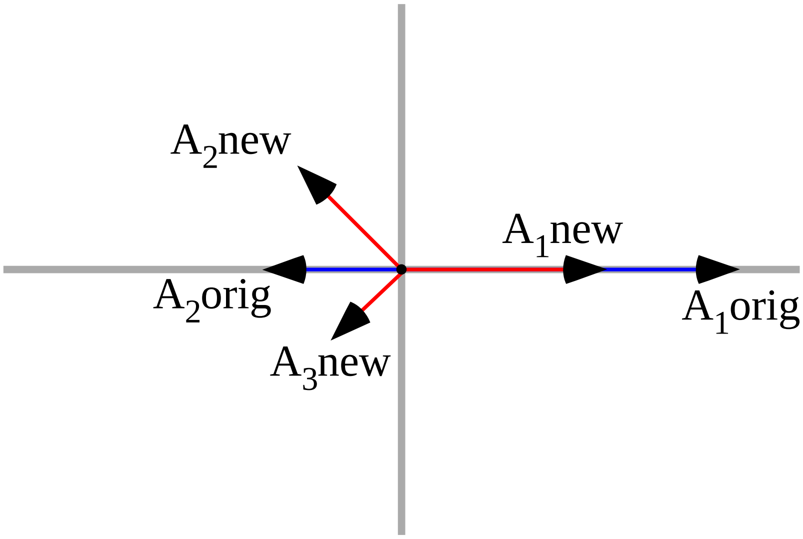


Figure 2.17: Vector Diagram Describing New Input Shaper.

(which normally occurs at time $t = 0$) depends upon the size of the step given to the closed-loop system. This step size depends upon the input shaper being used, as the original step size is scaled by the size of the first impulse. Therefore, it is desirable that the input shaper being used have a relatively small first step. Consequently, the new input shaper has variable initial impulse amplitude. This new input shaper can be seen in Figure 2.17. The original input shaper, whose impulses are denoted by $A_{\#}orig$, represents a standard ZV shaper. The new input shaper is created via a numerical routine to obtain zero residual vibration, an impulse summation of one and a specified size of the first impulse. Here, the first impulse is roughly 60% the size of a standard ZV shaper's first impulse. Also, as can be seen from Figure 2.17, this new input shaper requires a third impulse (labeled A_3new) to ensure complete vibration cancelation. Once the new input shaper is obtained, the size of the first impulse is re-entered into the concurrent design equations listed above to allow for further increase of the PD gains. The response result is shown in Figure 2.18 as "Concurrent Design #2".

The concurrently designed controller now provides arguably better disturbance rejection and a clearly superior step response than either of the sequentially designed

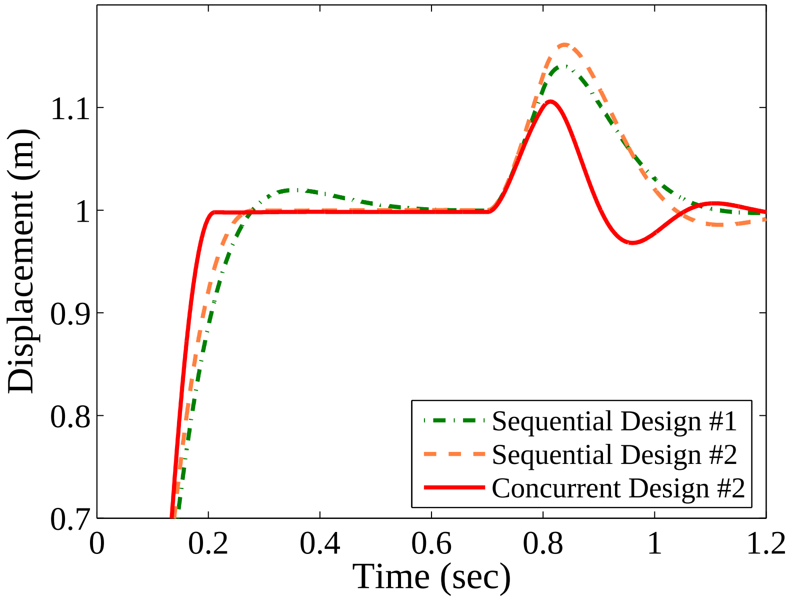


Figure 2.18: Redesigned Input Shaper in Concurrent Design.

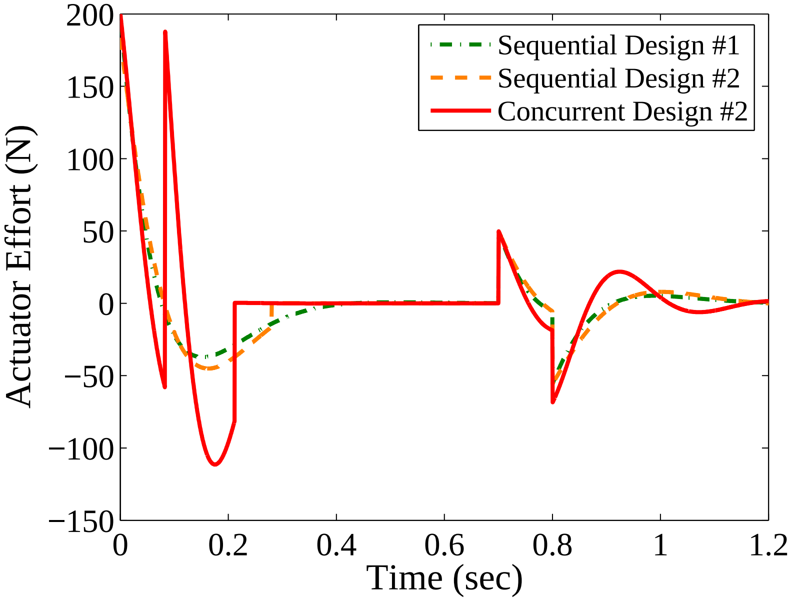


Figure 2.19: Actuator Requirements for System with Redesigned Input Shaper.

controllers. However, even though it does not violate the maximum actuator limit, this response clearly requires a more aggressive actuator effort, as shown in Figure 2.19.

All four designs are compared in Figures 2.20 and 2.21. Concurrent design one

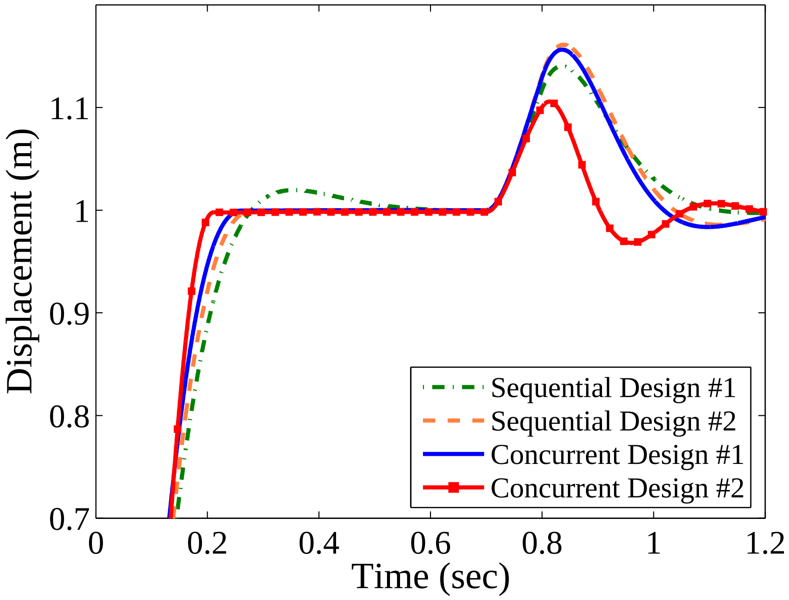


Figure 2.20: Full Comparison.

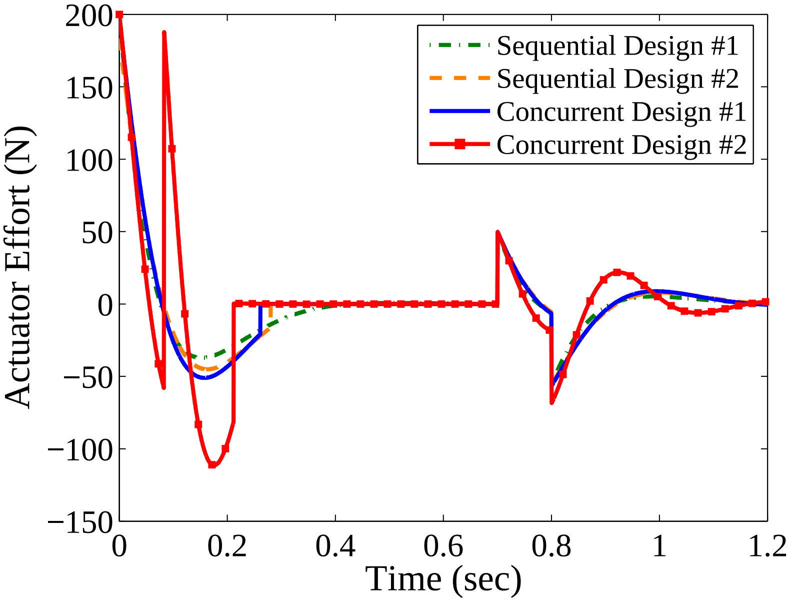


Figure 2.21: Full Comparison - Actuator Requirements.

utilizes standard ZV shapers, while concurrent design two utilizes the new, variable impulse shaper with reduced first impulse. Both concurrent designs outperform the sequential designs, while the two concurrent designs present a typical tradeoff between performance and actuator requirements. While each controller stayed within its maximum actuator limits, the controller developed by “Concurrent Design #2” clearly requires a more aggressive and demanding actuator response.

One final note that can be made in regards to Figure 2.20 is that these four control schemes indicate the presence of a spectrum that spans the gap between sequential design and concurrent design. The “Sequential Design #1” response was obtained by completely ignoring all effects of input shaping and utilizing only the feedback controller to minimize settling time. “Sequential Design #2” was actually a simple and incomplete version of a concurrent design in that it gave rise time some priority knowing that input shaping would eventually cancel vibration. “Concurrent Design #1” fully understood and used the knowledge of standard input shaping to design a superior control scheme. Finally, “Concurrent Design #2” utilized a deeper understanding of how input shapers and feedback controllers interact to develop new input shapers that yield faster responses.

2.4 Look-up Tables for Varying Parameters

The concurrent design scheme presented thus far yields results that are specific to the plant's mass and the desired step size. However, many real systems are required to move varying masses over varying distances. For example, a pick-and-place robot may need to move three different masses over three different distances during its normal operation. For optimal performance, this requires multiple PD and shaper parameters. While simple design schemes could be performed in real time, this section demonstrates the use of a look-up table.

To demonstrate the solution's dependence on the mass and step size, several examples will be discussed. Figure 2.22 shows the step response to the concurrently designed control scheme when both the design step size, L_{des}, and the actual step size, L_{act}, are equal to unity. When the same controller parameters ($L_{des} = 1$) are

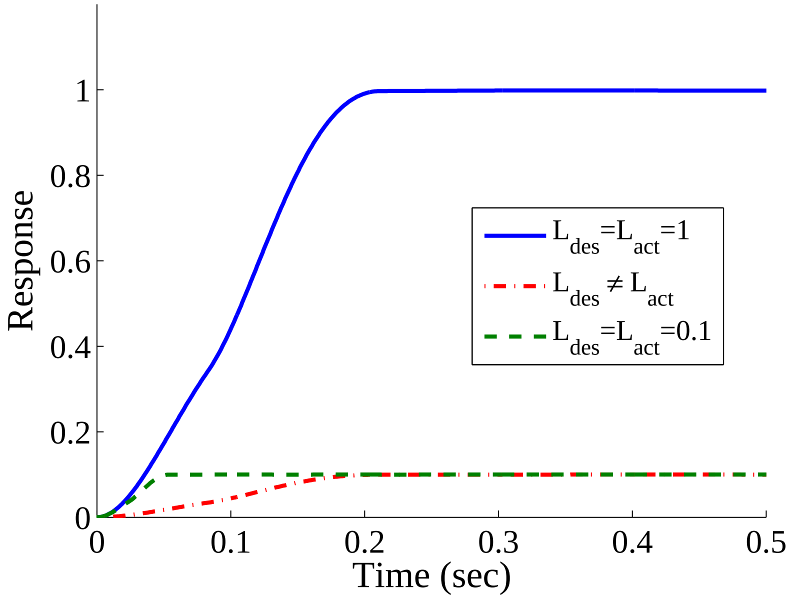


Figure 2.22: Varying Step Size (L) Responses.

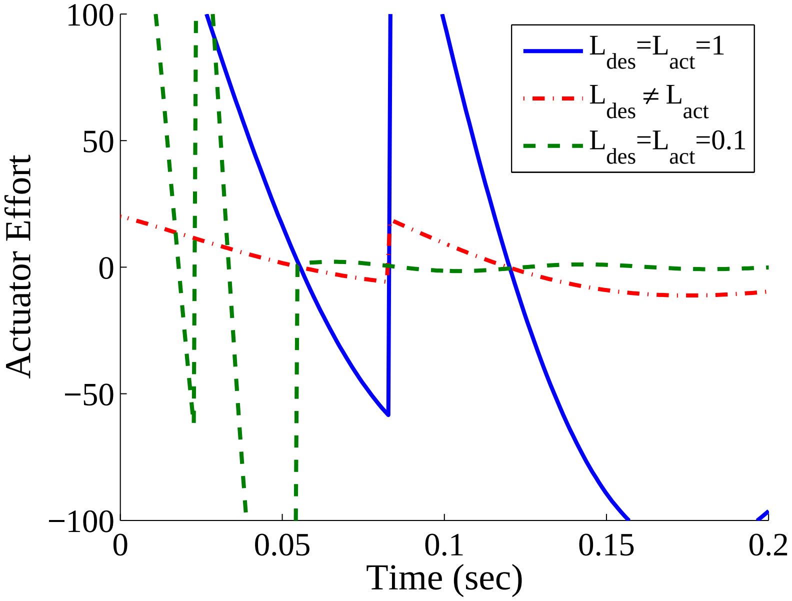


Figure 2.23: Actuator Efforts Required for Varying Step Size (L) Responses.

used to move the system one-tenth ($L_{act} = 0.1$) the designed move distance, the result is much slower than if both the designed and actual move distances were 0.1 ("$L_{des} = L_{act} = 0.1$"). The reason for this dependance on L is seen by the actuator efforts shown in Figure 2.23. When the actual step size, L_{act}, is lower than the

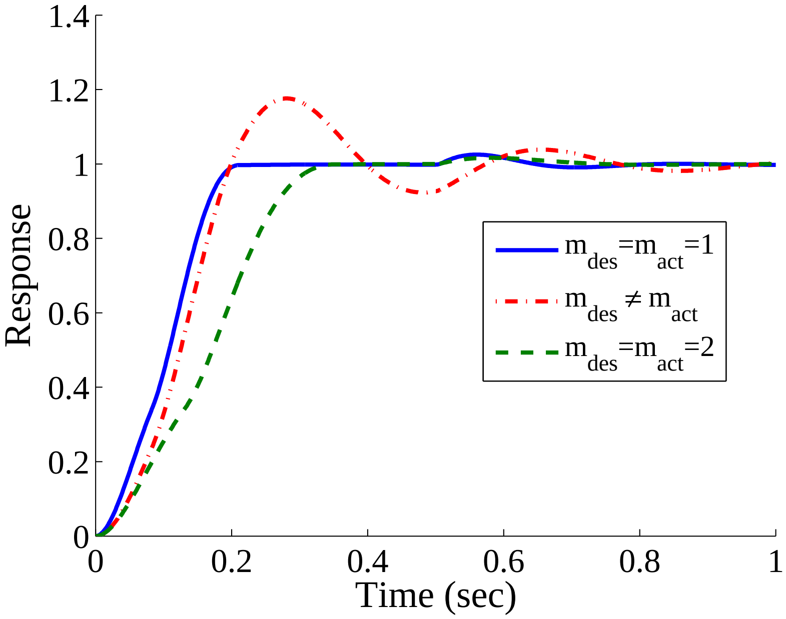


Figure 2.24: Varying Plant Mass Responses.

designed step size, L_{des}, the system under utilizes its actuator, resulting in slower movements.

If the step sizes are kept the same, but the designed and actual plant masses are varied, then a modeling error situation arises. Remember from Section 2.3 that the plant mass affects the closed-loop natural frequency. Therefore, if the designed and actual plant masses are equal, a fast, vibration free motion will occur. This can be seen in Figure 2.24 where $m_{des} = 1$ and $m_{act} = 1$. However, if m_{des} is left at unity while the actual plant mass, m_{act}, is raised to 2, the actual closed-loop natural frequency is different from the natural frequency estimate used to design the input shaper. This results in non-zero residual vibration. Also note from Figure 2.25 that a higher m_{act} results in actuator limit violations ($U_{max} = 200$ here). If m_{des} is now adjusted to meet the actual mass of 2 units, the system is again vibration free (Figure 2.24) and within the actuator limits (Figure 2.25).

Figures 2.26 - 2.32 show each control parameter's dependance on mass and L. This information can be used to establish look-up tables for a real-world system.

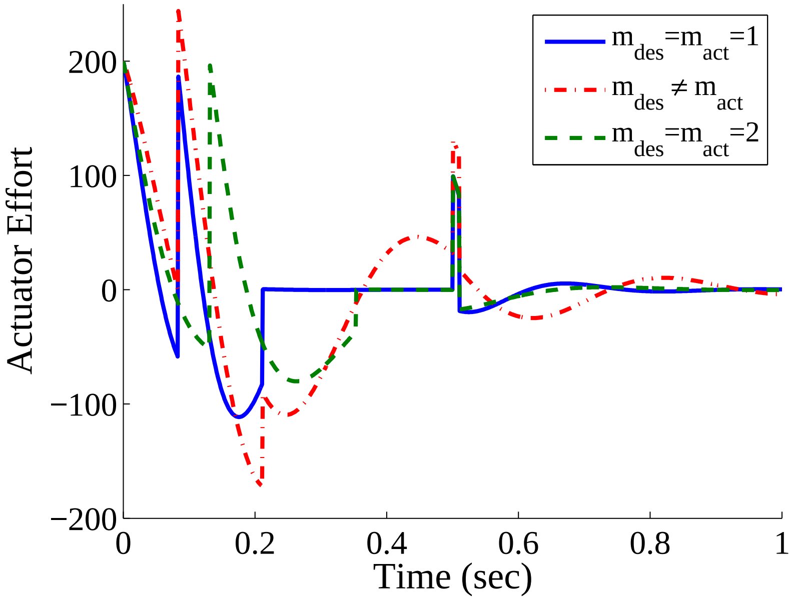


Figure 2.25: Actuator Efforts Required for Varying Plant Mass Responses.

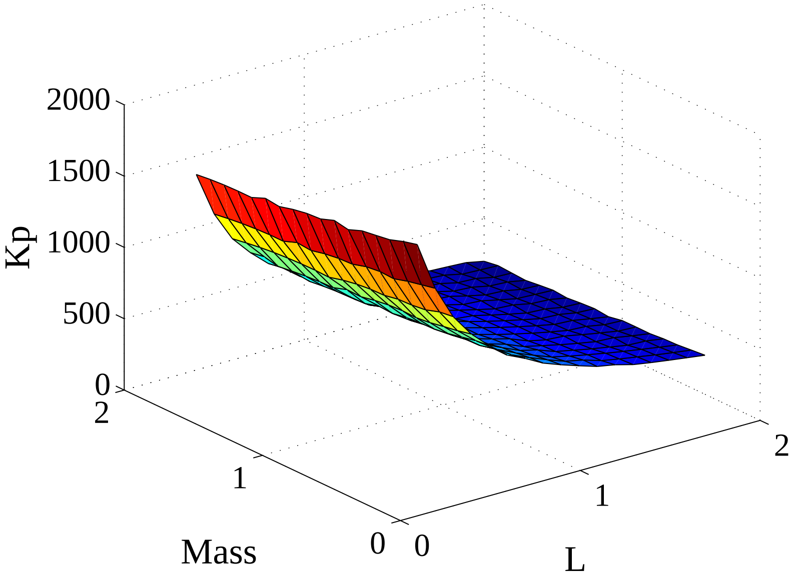


Figure 2.26: Optimal K_p Versus Mass and Move Distance.

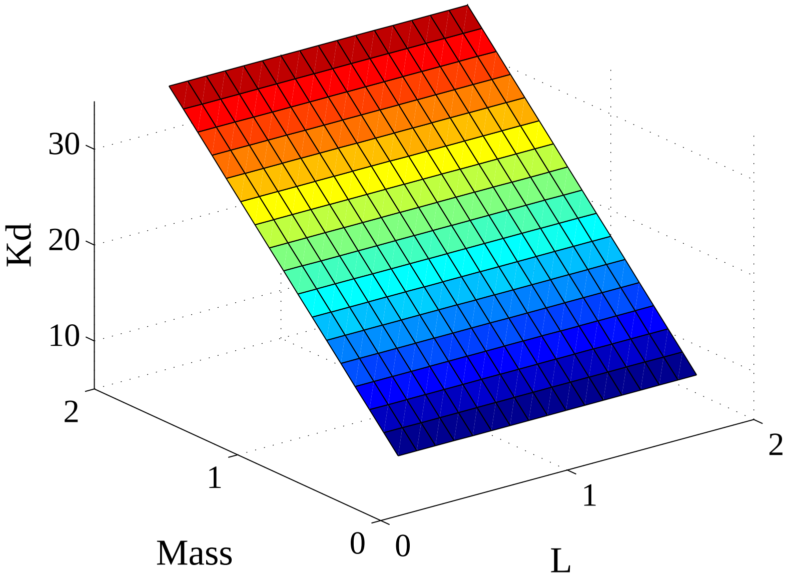


Figure 2.27: Optimal K_d Versus Mass and Move Distance.

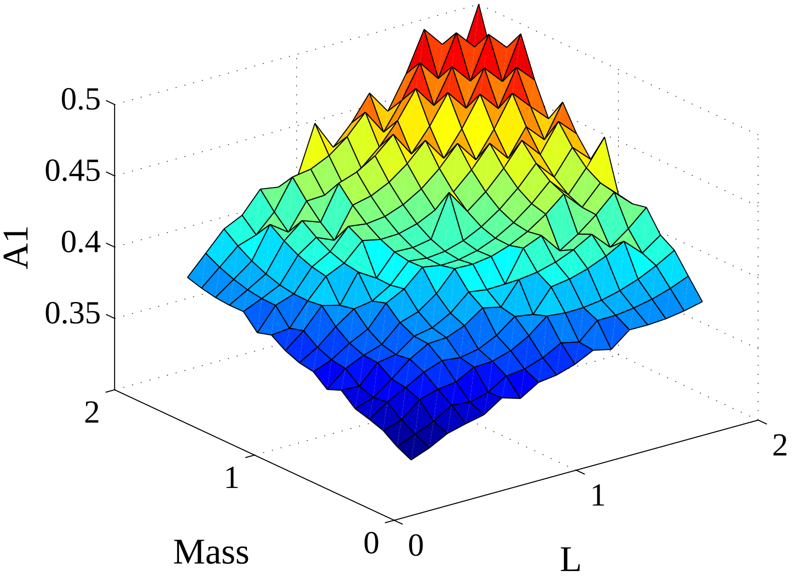


Figure 2.28: Optimal A_1 Versus Mass and Move Distance.

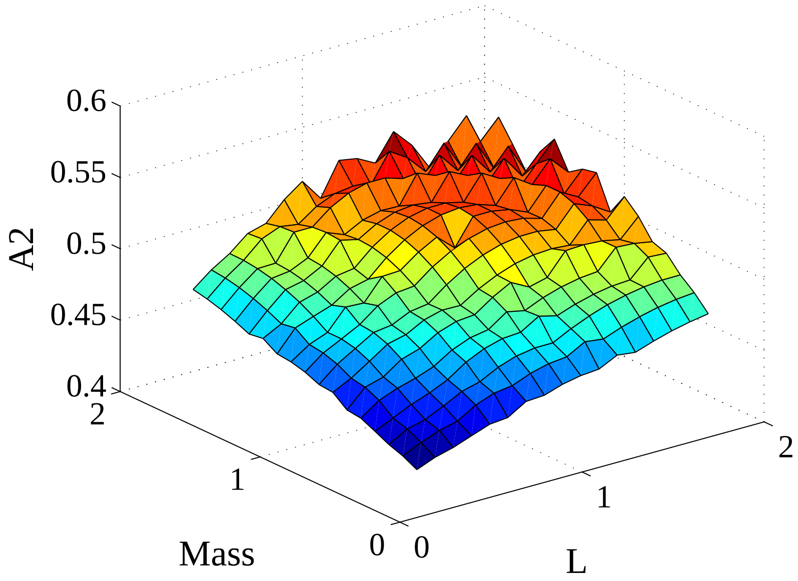


Figure 2.29: Optimal A_2 Versus Mass and Move Distance.

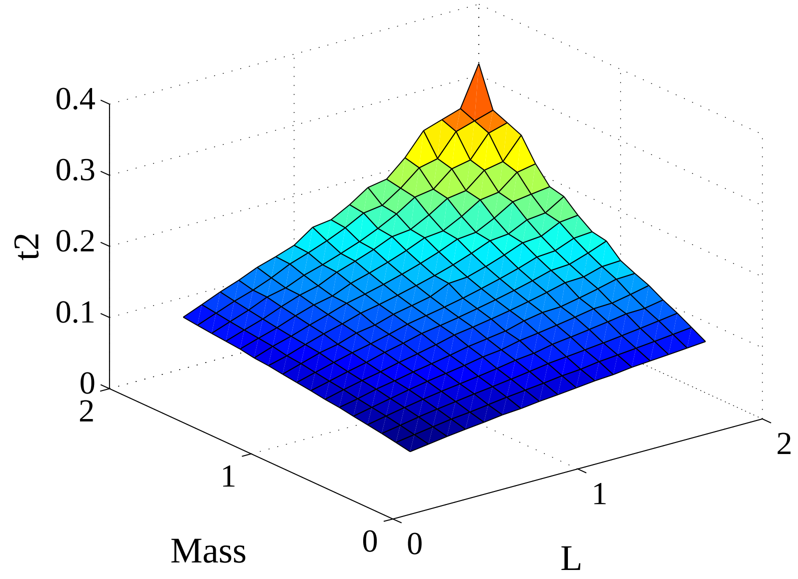


Figure 2.30: Optimal t_2 Versus Mass and Move Distance.

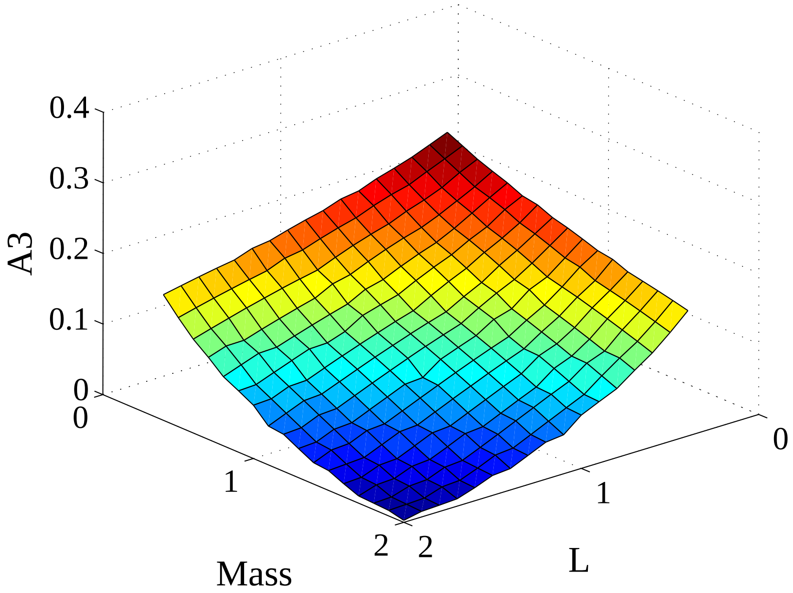


Figure 2.31: Optimal A_3 Versus Mass and Move Distance.

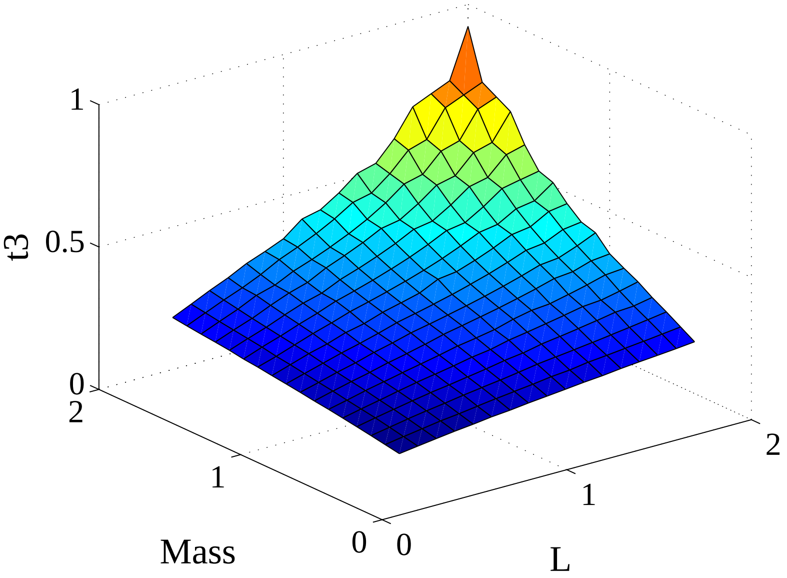


Figure 2.32: Optimal t_3 Versus Mass and Move Distance.

CHAPTER III

EXPERIMENTAL RESULTS USING A MASS PLANT

In order to verify the theoretical developments presented in the previous chapter, the superiority of concurrently designing outside-the-loop input shaping and PD feedback controllers was experimentally tested. As in the previous chapter, the plant under control was a mass, although the experimental setups did have some nonlinear effects such as friction which are not modeled in the controller design procedures. There were four main experiments completed, two on a rotational inertia and two on a translating mass.

3.1 Rotational Mass Experiments

The rotating inertia experiments were completed on a setup similar to the one depicted in Figure 3.1. The setup consisted of a rotational motor and a rotational inertia (mass added to the motor shaft). The inertia was determined by forcing the

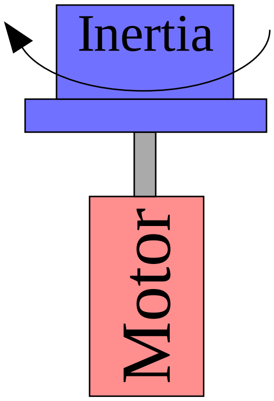


Figure 3.1: Experimental Setup.

motor to output various constant torque values and then measuring the resultant angular acceleration. The inertia value calculated in this manner was then checked by estimating the inertia of the added mass via its density and physical dimensions and finding the inertia of the motor from the motor's documentation.

3.1.1 Rotational System Verification

Because the experimental setup is a real system, it will not behave in a completely linear manner. For example, this motor has some kinetic friction that affects the rotational motion. In order to verify that the system would behave in an approximately linear manner, several system verification trials were completed.

These system verification trials looked at the motor's step response to various proportional (K_p) and derivative (K_d) gains. Note that the derivative action was implemented as velocity feedback. A nonlinear, least-squares algorithm was then used in MATLAB to fit linear mass-under-PD-control responses to the previously obtained experimental responses. Since the rotational inertia had already been determined, the least-squares algorithm chose the appropriate proportional and derivative gains which minimized the error between the linear, simulated step response and the experimentally obtained step response. The relationship between the K_p and K_d gains used in the experiments to those found from the MATLAB least-squares algorithm was used to verify that the motor-mass system was acting in a near-linear manner. The experimentally-used and simulation-determined proportional and derivative gains are summarized in Table 3.1. Here, "Exp" refers to the values entered into the PLC controlling the motor and "Sim" refers to the values determined by the least-squares algorithm in MATLAB. Note that both the proportional (K_p) and derivative (K_d) gains are consistent between experiment and linear simulation.

Figures 3.2 through 3.11 show the experimentally measured step responses, the simulation based responses and the gains determined by the least-squares algorithm.

Table 3.1: Linearity Verification Results - Rotational Experiments

$Exp\ \ K_p$	$Sim\ \ K_p$	$Exp\ \ K_d$	$Sim\ \ K_d$
0.35	0.37	0.006	0.004
0.35	0.37	0.008	0.006
0.35	0.37	0.010	0.008
0.35	0.37	0.012	0.011
0.35	0.38	0.014	0.013
0.127	0.13	0.002	0.002
0.127	0.129	0.004	0.004
0.127	0.129	0.006	0.006
0.127	0.129	0.008	0.009
0.127	0.13	0.010	0.010

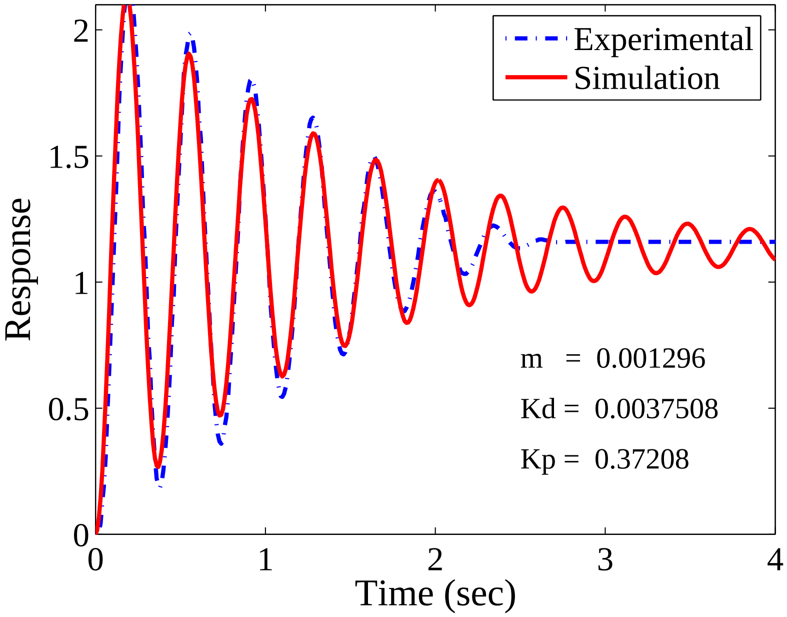


Figure 3.2: Mass Calibration Experiment 1.

The first half of the experiments, Figures 3.2 through 3.6, have the same experimental, proportional gain (Exp $K_p = 0.35$). Figure 3.2 begins with an experimental, derivative gain of 0.006. The next four experiments have the same experimental, proportional gain and an incrementally (increment is 0.002 units) higher derivative gain. The same is true for Figures 3.7 through 3.11. Here, the experimental, proportional gain is held fixed at (Exp $K_p = 0.127$) and the experimental, derivative gain rises incrementally from 0.002 to 0.010.

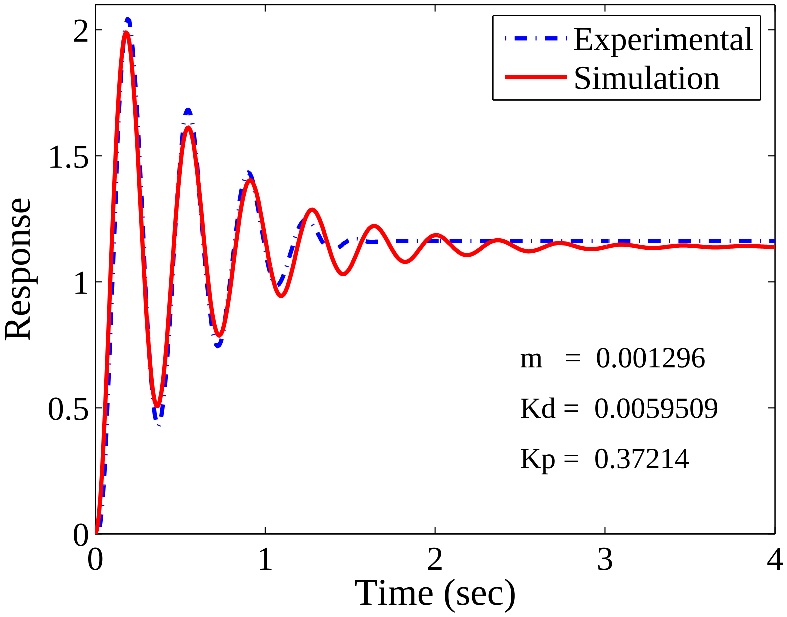


Figure 3.3: Mass Calibration Experiment 2.

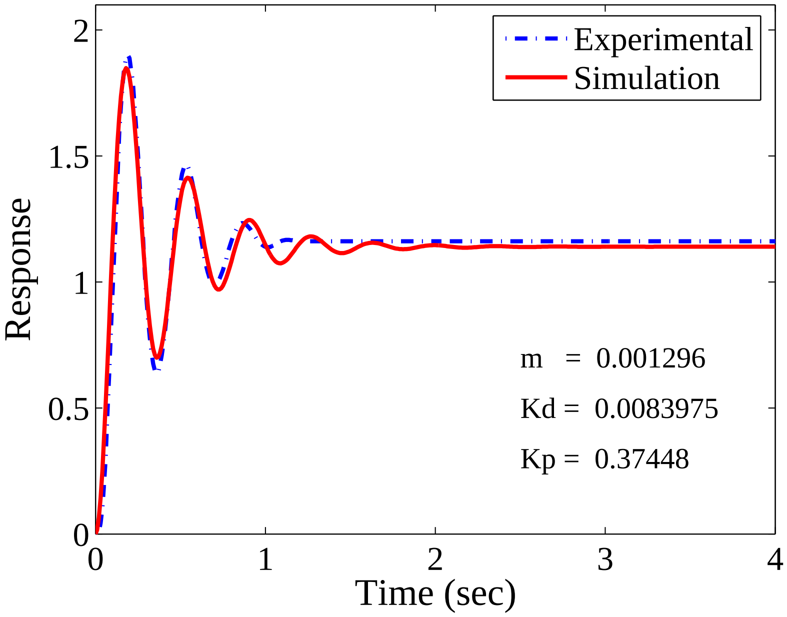


Figure 3.4: Mass Calibration Experiment 3.

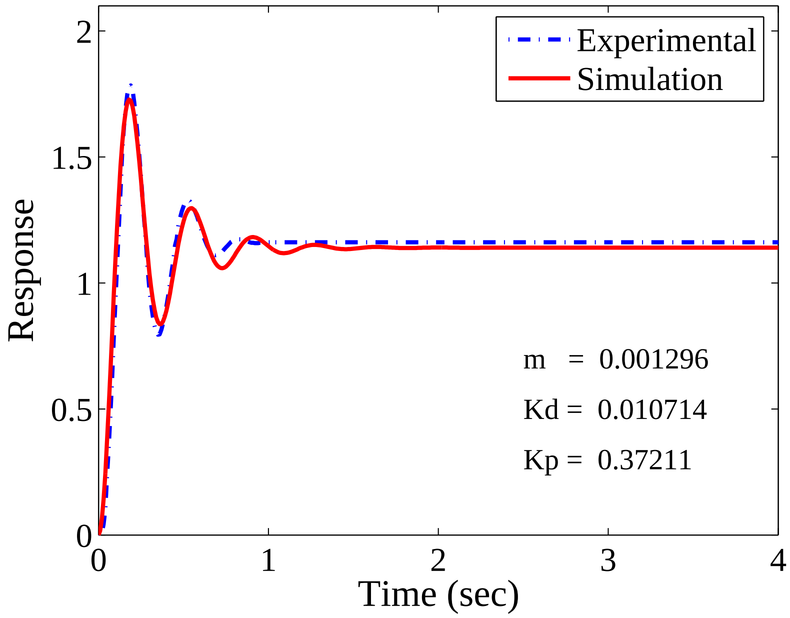


Figure 3.5: Mass Calibration Experiment 4.

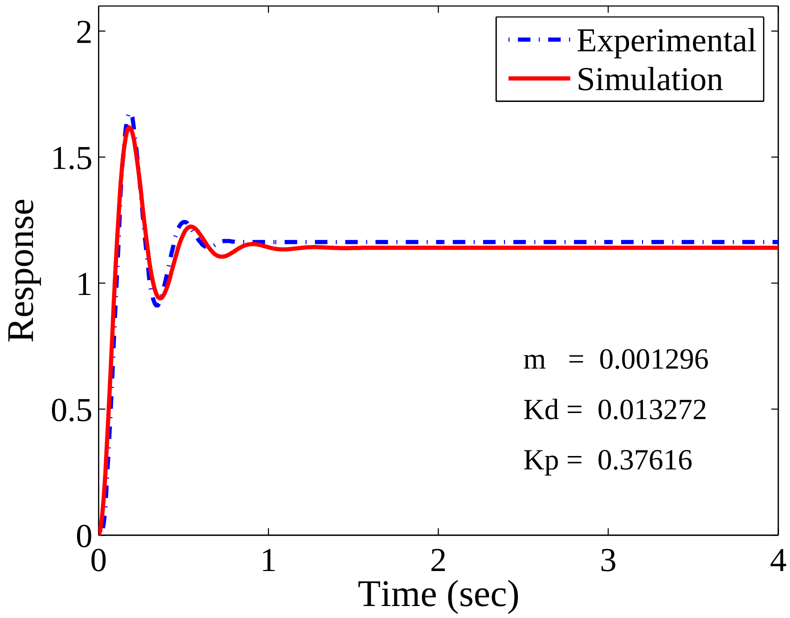


Figure 3.6: Mass Calibration Experiment 5.

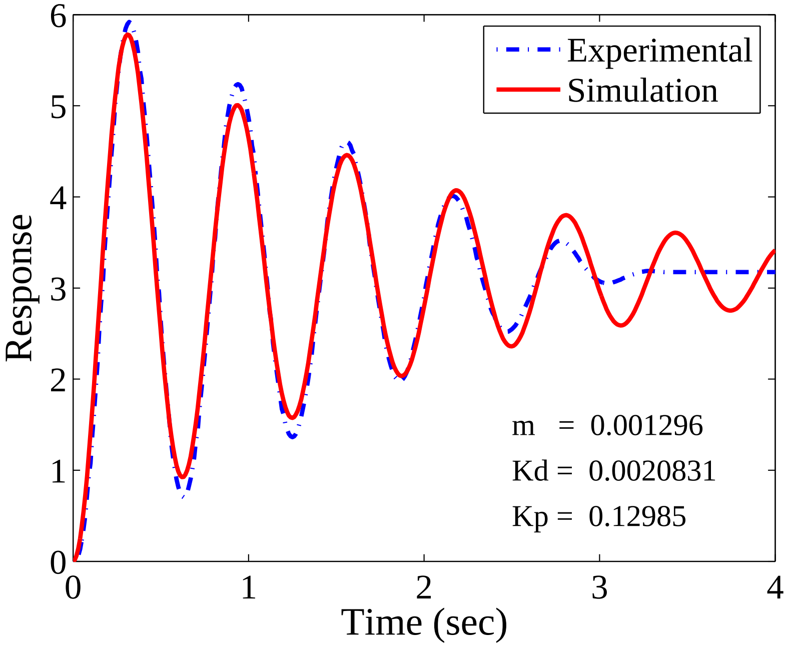


Figure 3.7: Mass Calibration Experiment 6.

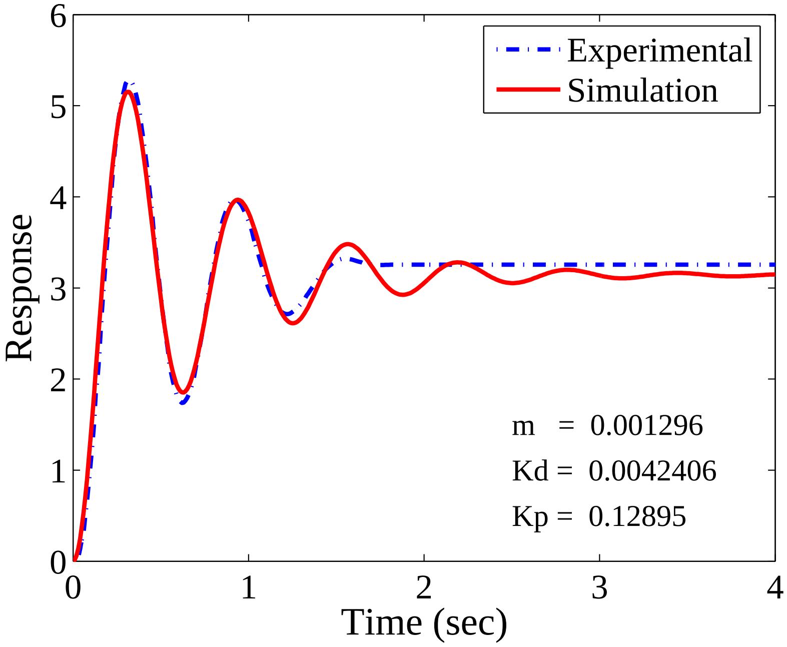


Figure 3.8: Mass Calibration Experiment 7.

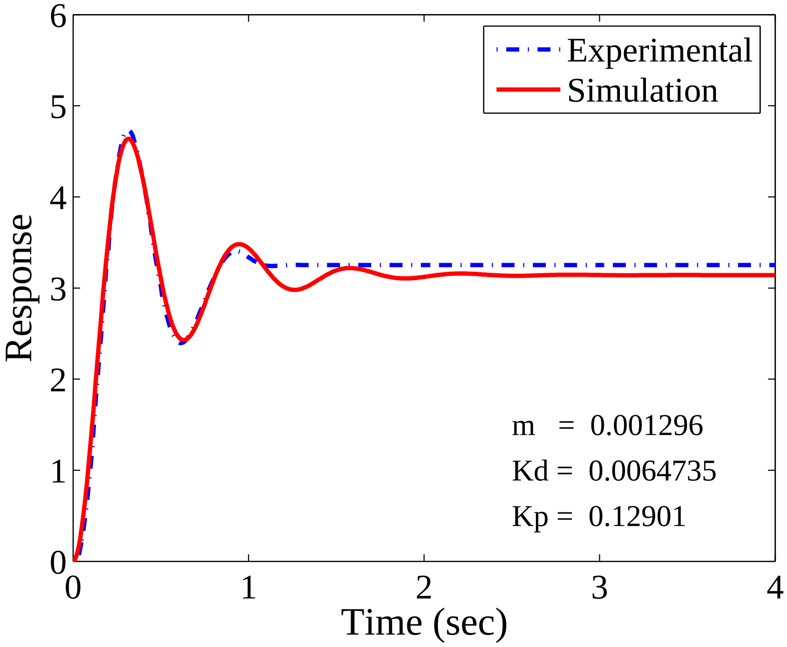


Figure 3.9: Mass Calibration Experiment 8.

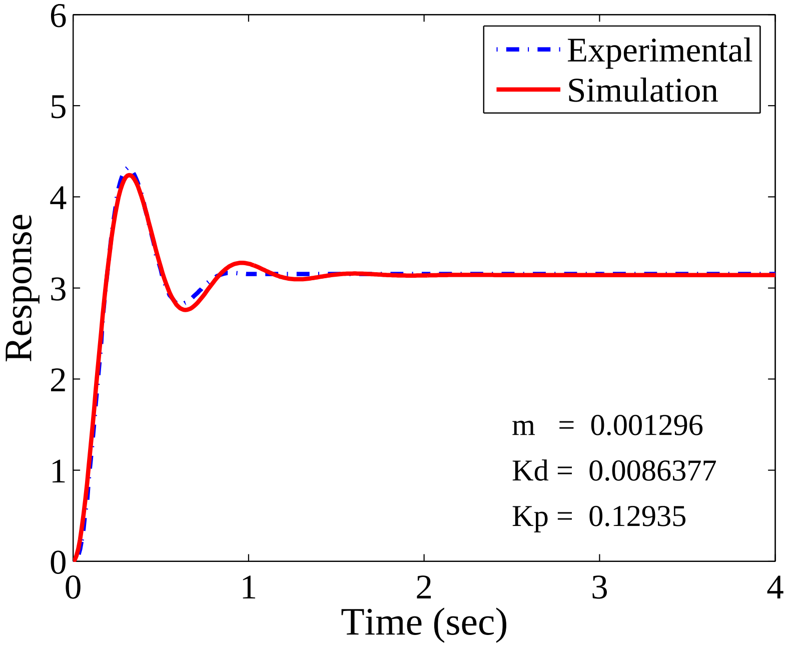


Figure 3.10: Mass Calibration Experiment 9.

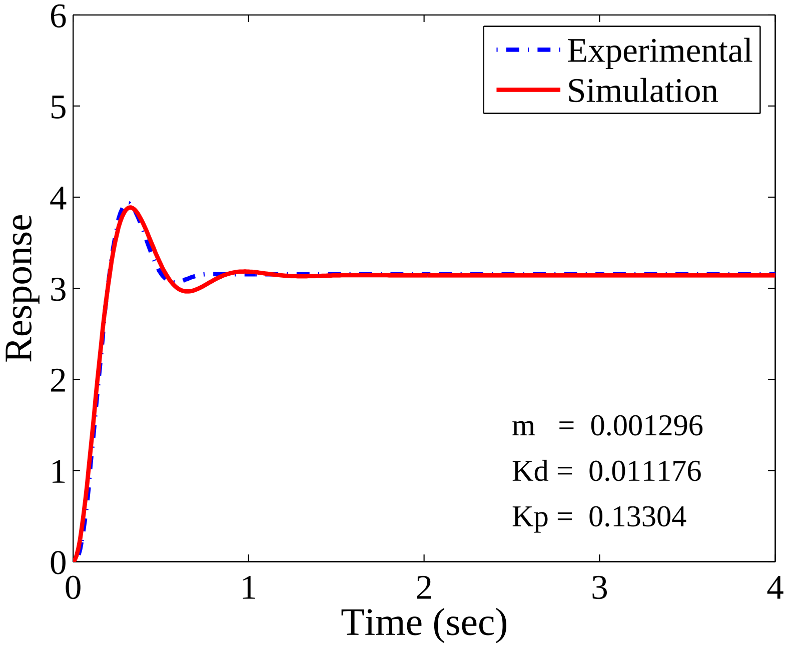


Figure 3.11: Mass Calibration Experiment 10.

3.1.2 Rotational Mass: Experiment Set #1

The system was commanded to move π radians with a maximum actuator effort of $0.4Nm$. The design parameters sought to achieve the fastest 2% settling time while maintaining less than 5% overshoot and a time constant $(\frac{1}{\zeta\omega_n})$ less than, or equal to, $0.15sec$.

Once the experimental setup's behavior was understood, sequential and concurrent design procedures similar those described in Section 2.3 were performed with the design specifications that were detailed in the previous paragraph. This design procedure chose the appropriate PD gains and shaper parameters for both the sequentially and concurrently designed controllers. And, like the simulation results in Section 2.3, this experiment sought to give the sequential design method the greatest chance for success. For example, simply looking at a mass under PD control, and desiring the smallest settling time, one would naturally choose a damping ratio between 0.7 and 0.8. However, the sequential design procedure used here gave some importance to rise time (although it was not technically a design parameter). This was done so

Table 3.2: Rotational Mass Design #1

$SequentialDesign$	$ConcurrentDesign$
$K_p = 0.127$	$K_p = 0.33$
$K_d = 0.0177$	$K_d = 0.017$
$A_1 = 1$	$A_1 = 0.38$
$A_2 = N/A$	$A_2 = 0.48$
$A_3 = N/A$	$A_3 = 0.14$
$t_2 = N/A$	$t_2 = 0.12$
$t_3 = N/A$	$t_3 = 0.312$
$\omega_n = 9.9$	$\omega_n = 16.51$
$\zeta = 0.69$	$\zeta = 0.4$

that the sequentially designed control scheme would be faster (when an input shaper or natural friction was added) than it would have been had rise time been ignored. The fact that the concurrently designed scheme still outperformed the sequentially designed scheme under these circumstances shows all the more that it is a generally superior design strategy.

Using the design specifications noted at the beginning of this section, the gains and shaper parameters shown in Table 3.2 were found using the sequential and concurrent design procedures. There are several things worth noting from Table 3.2. First, as would be expected, the concurrent design yields a significantly higher K_p value. However, somewhat unexpected is the virtual equality between the sequentially and concurrently determined K_d values. This did not occur in the simulations described in Section 2.3. This phenomenon occurred here because the disturbance rejection constraint (time constant, t_c) was equally as “stringent” as the overshoot constraint. The sequential design technique required the indicated K_d value to meet both its overshoot and time constant constraints. The concurrent design uses input shaping to meet its overshoot constraint, but must still rely on K_d to meet its time constant constraint. This was intentionally done to show one of the intricacies of the concurrent design process. Some constraints, like disturbance rejection, are not directly affected by the presence of an outside-the-loop input shaper. They are only affected by the closed-loop system characteristics. Therefore, if a system's set of performance constraints heavily restrict the closed-loop system's form or characteristics, then the sequential and concurrent design techniques will begin to yield similar results. However, it is clear from Table 3.2, that enough freedom was given to the concurrent design technique that a much higher K_p was chosen, resulting in an overall higher closed-loop frequency (ω_n) and lower damping (ζ).

It should also be noted that the sequentially designed control system was subsequently given no input shaper. This occurred because the closed-loop system had such a high damping ratio. Even though a damping ratio of 0.69 will yield non-zero overshoot, the system friction eliminated this undesired aspect and actually helped the unshaped, sequentially designed system to settle faster than it would have without friction. Note that by not using an input shaper and allowing friction to help the sequentially designed system's response, a very favorable sequentially designed response will be used to determine if concurrent design procedures yield any improvement. Figure 3.12 shows the simulated and experimental responses of the closed-loop system that was sequentially designed.

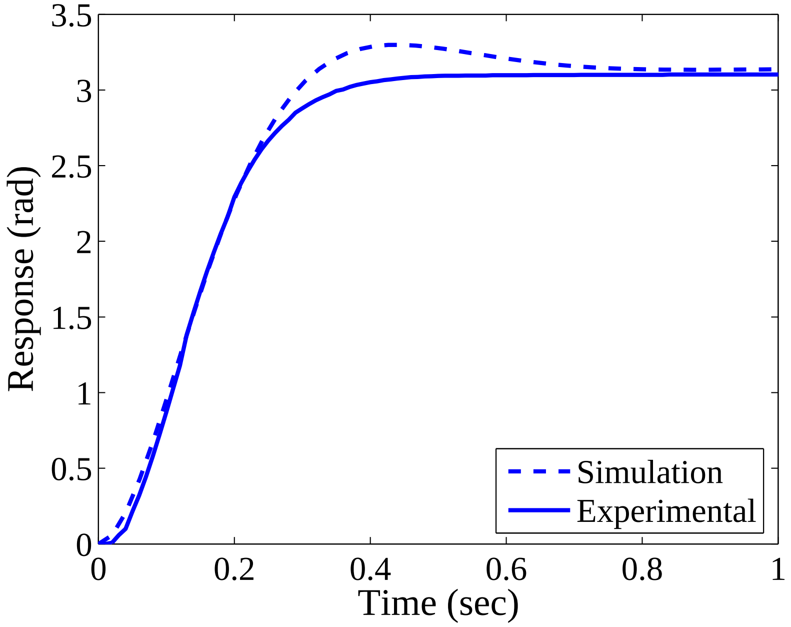


Figure 3.12: Sequentially Designed System Response.

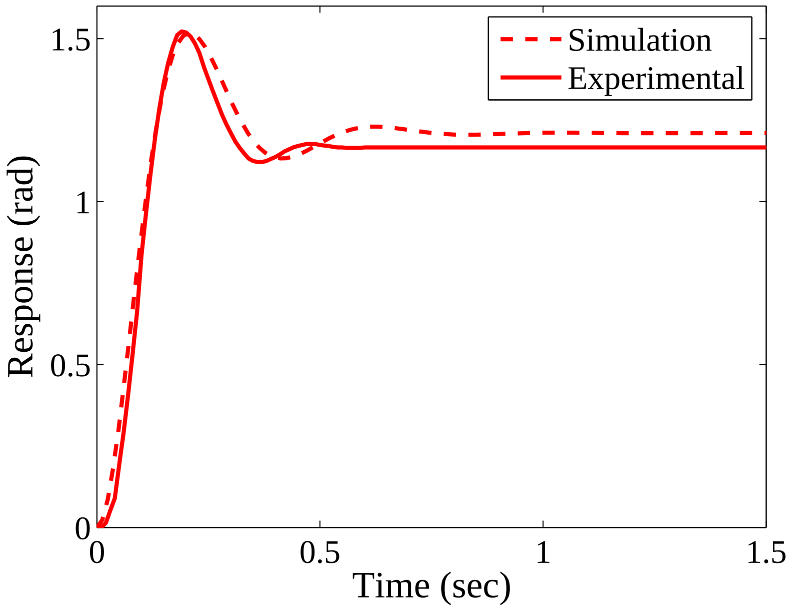


Figure 3.13: Concurrent Design Calibration.

Figure 3.13 shows an unshaped step response of the closed-loop (PD+Mass) system designed via the concurrent design scheme. This was done so that minor adjustments to the P and D gains could be made in the PLC so that the closed-loop experimental response closely matched the simulated closed-loop response (the simulated and actual P and D gains are slightly different due to modeling errors and nonlinearities). Remember that the concurrently designed control system is relatively non-oscillatory due to the stringent disturbance rejection constraint used in this particular example.

With the concurrently designed P and D gains found, the input shaped response (using the input shaper specifically designed along with the P and D gains) is shown in Figure 3.14. Here, the "Simulated" response is that predicted by the concurrent design algorithm. Using the experimentally adjusted PD gains (Figure 3.13) and the concurrently designed input shaper, the response labeled "Original Design" is obtained. This is obviously more oscillatory than expected. A zoomed in view of Figure 3.14 (shown in Figure 3.15) indicates that the "Original Design" peaks before the final impulse time of 0.31 seconds. A manual adjustment is made here on the

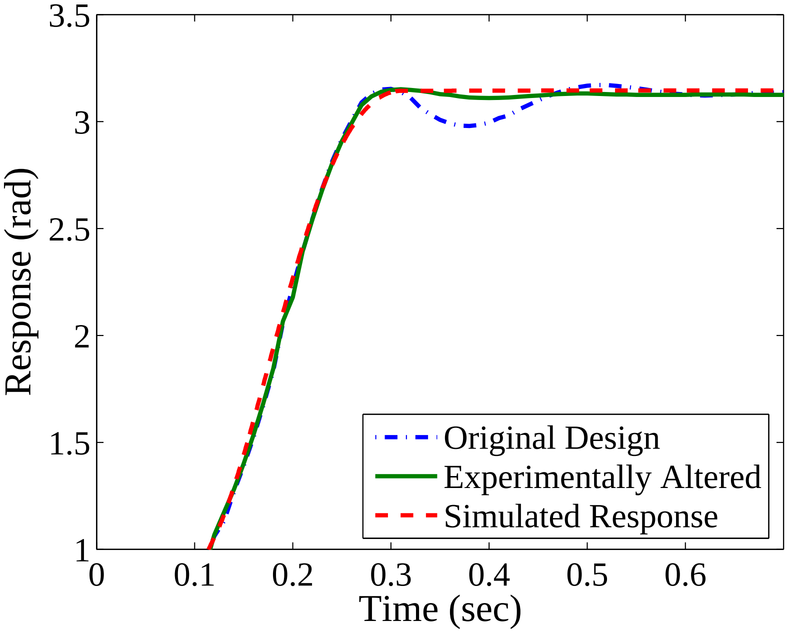


Figure 3.14: Concurrently Designed System Response.

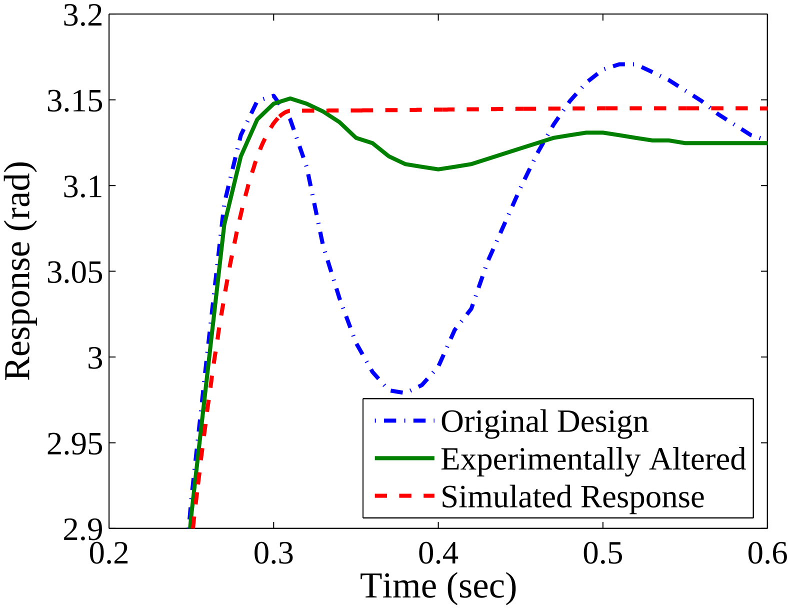


Figure 3.15: Concurrently Designed System Response - Zoomed-in View.

final shaper time (its dropped from 3.1 sec to 2.9 sec) to produce the well behaved "Experimentally Altered" response. This shaper error is most likely due to modeling errors and nonlinearities. This manual adjustment of the input shaper is a reasonable procedure to expect on many real-world systems.

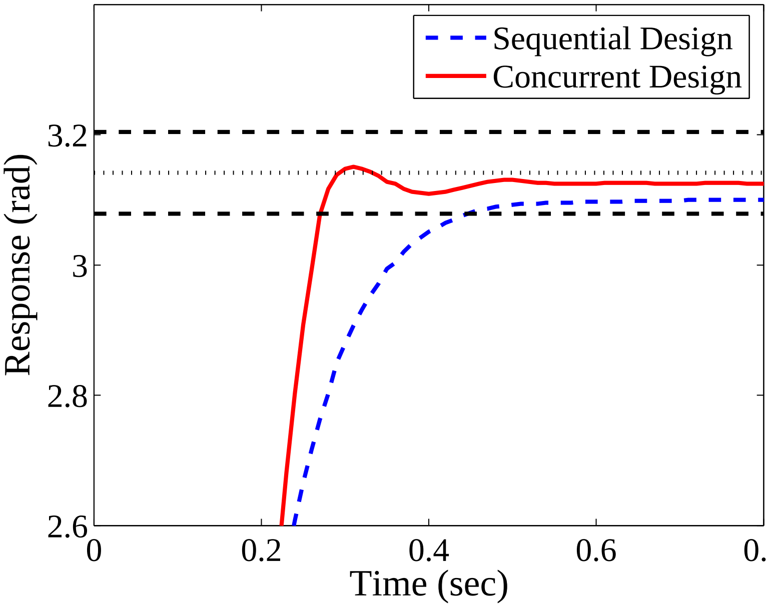


Figure 3.16: Design Response Comparison.

Figure 3.16 compares the sequentially designed and concurrently designed system responses. The dotted black line indicates the desired final position, whereas the dashed black lines indicate the 2% settling time boundaries. Clearly, the concurrently designed control system has a faster rise time and 2% settling time ($\sim$ 40% faster in terms of t_s), as well as less steady state error (due to the higher P gain). However, it should be noted that the undershoot present in the concurrent design scheme is sometimes larger than indicated here (it varied between trials), sometimes dipping temporarily below the 2% settling time line. Regardless, the concurrent design scheme is still faster and has less steady-state error. Remember also that this comparison somewhat favored the sequential design scheme - so any improvement over it is all the more notable.

Figure 3.17 shows the actuator efforts required for the responses shown in Figure 3.16. The lines without data markers indicate the simulated actuator requirements, whereas the lines with data markers indicate the experimentally measured actuator requirements. The blue, dashed lines are for the sequentially designed system ("Seq.

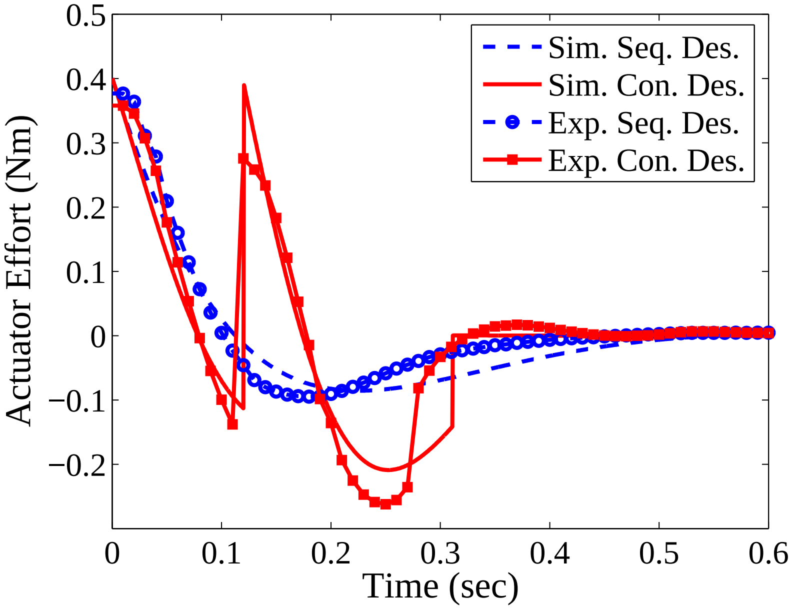


Figure 3.17: Required Actuator Efforts.

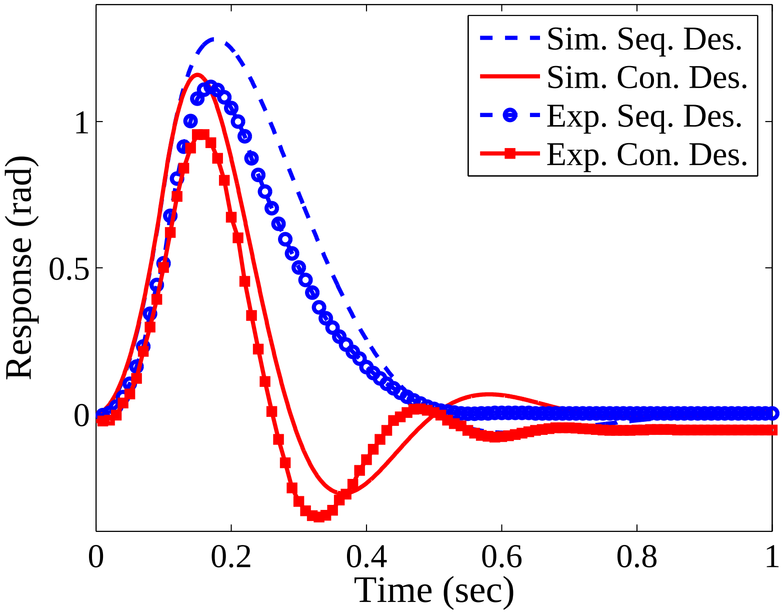


Figure 3.18: Disturbance Rejection Capabilities.

Des."), and the red, solid lines are for the concurrently designed system ("Con. Des.").

Figure 3.18 shows the predicted and actual disturbance responses for both the sequentially designed ("Seq. Des.") and concurrently designed ("Con. Des.") systems.

Table 3.3: Rotational Mass Design #2

$Sequential Design$	$Concurrent Design$
$K_p = 0.127$	$K_p = 0.375$
$K_d = 0.0117$	$K_d = 0.0086$
$A_1 = 0.83$	$A_1 = 0.34$
$A_2 = 0.17$	$A_2 = 0.41$
$t_2 = 0.356$	$A_3 = 0.25$
	$t_2 = 0.111$
	$t_3 = 0.265$
$\omega_n = 9.9$	$\omega_n = 17.51$
$\zeta = 0.46$	$\zeta = 0.19$

The error between the predicted and actual responses is believed to result from unmodeled friction helping to reject the disturbance. However, note that each response closely matches the predicted form and timing.

3.1.3 Rotational Mass: Experiment Set #2

A second set of experiments was performed using the rotational mass. The mass was again commanded to rotate π radians with a maximum actuator effort of $0.4Nm$. However, the overshoot allowance was increased to 20% and the time constant constraint was increased to $0.3sec$. These performance specifications and constraints led to the sequential and concurrent design parameters shown in Table 3.3.

To ensure that the actual mass would respond as expected, a few calibration experiments were performed. These calibrations were performed by finding the unit step response (experimentally and in simulation) under the PD gains shown in Table 3.3. The PD gains used in the PLC were adjusted so that the actual system would respond as predicted in simulation. Using the sequentially designed PD gains, the step response of the rotational inertia is shown in Figure 3.19. Note that the unmodeled friction caused the experimental response to appear more damped than was predicted. This actually aids the sequentially designed controller to improve its settling time without violating any constraints, so the sequentially designed feedback controller used in the actual experiments was the same as was designed in MATLAB

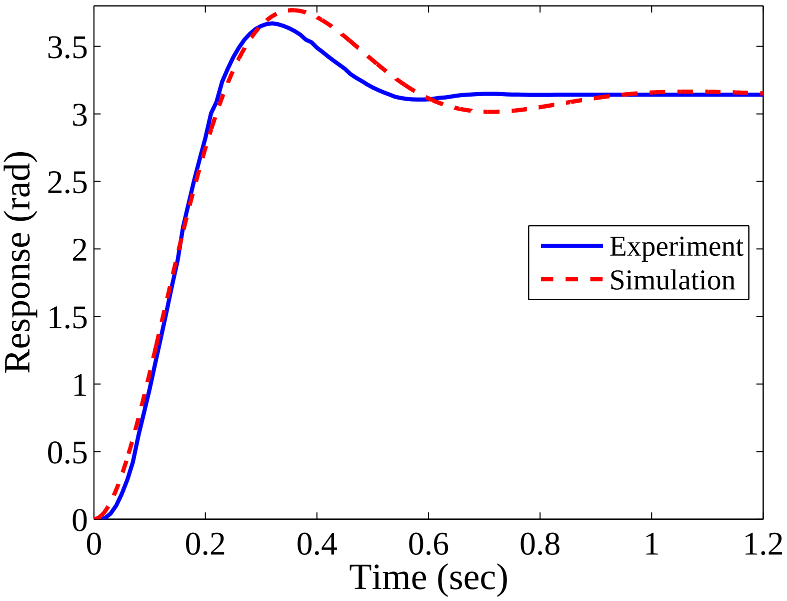


Figure 3.19: Sequential Design Calibration.

and presented in Table 3.3. However, since the closed-loop system peaks earlier than predicted, the time of the final shaper impulse was lowered from $t_2 = 0.356sec$ to $t_2 = 0.32sec$.

The same calibration procedure was performed for the concurrently designed controller. In this case, it was found that using the concurrently designed PD gains in the PLC resulted in a step response that was much more oscillatory than predicted. Therefore, the PD gains shown in Table 3.3 were adjusted from $K_p = 0.375$ and $K_d = 0.0086$ to $K_p = 0.3$ and $K_d = 0.011$. This adjustment made the actual feedback controller behave like the concurrently designed and simulated feedback controller, as is shown in Figure 3.20.

Once the two feedback systems (sequentially and concurrently designed) were behaving as designed, the respective input shapers were applied to test the overall control systems. Figure 3.21 compares the step responses of the sequentially and concurrently designed controllers. Here, the concurrently designed control scheme has a settling time that is 37.4% faster than the sequentially designed controller. In addition to improving upon the settling time, the concurrently designed controller has

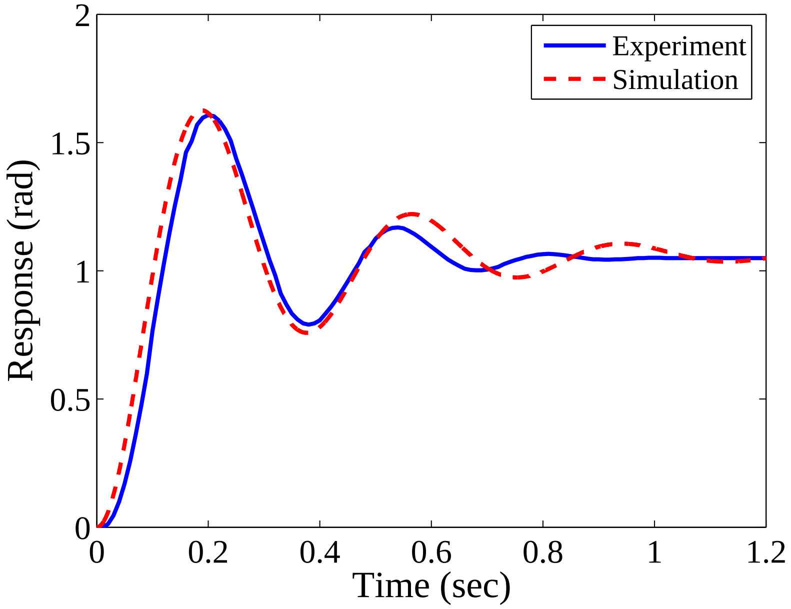


Figure 3.20: Concurrent Design Calibration.

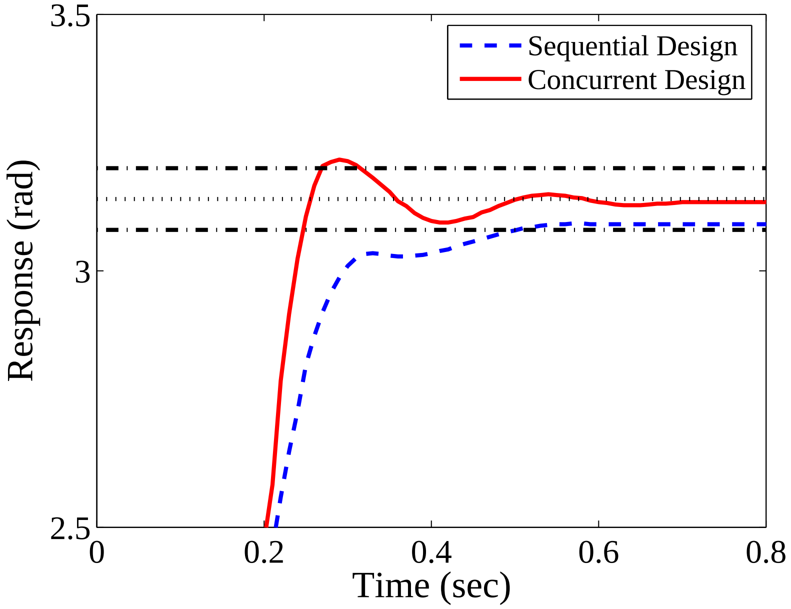


Figure 3.21: Step Response Comparison #1.

an 87.6% improvement in steady-state error. Note that this figure denotes the desired position of π *rad* by a dotted line and the settling time boundaries by dash-dot lines. Figure 3.22 shows the actuator effort requirements for the step responses shown in Figure 3.21. Neither control scheme violates the actuator limits, but the concurrently designed controller does require a more aggressive actuator response.

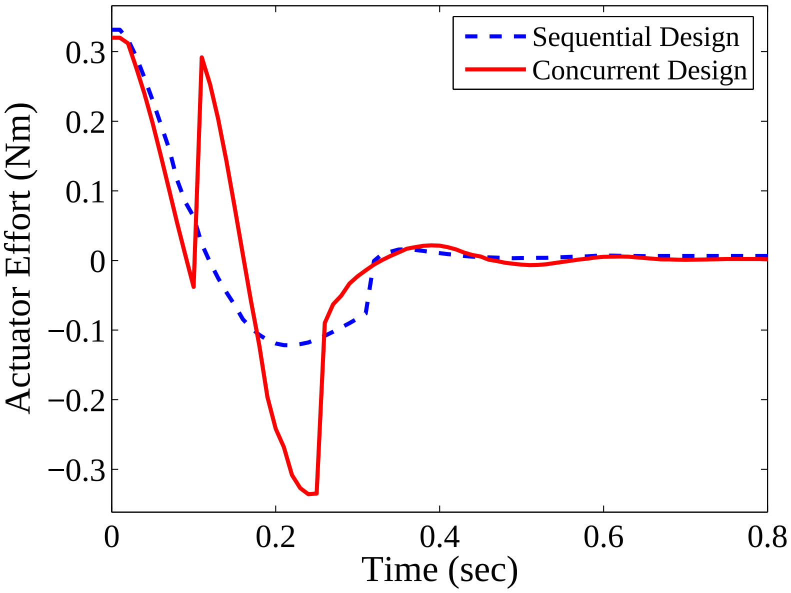


Figure 3.22: Actuator Comparison.

A second response comparison was performed after the input shapers were slightly adjusted to improve settling time. Because the sequentially designed controller peaked below the settling time window, its input shaper was adjusted to have a slightly higher first impulse (A_1). Then, because the concurrently designed controller also peaked outside the settling time window, its impulse amplitudes were also adjusted. The new controllers had step responses as shown in Figure 3.23. Note that each controller shown here had a faster settling time than the respective control scheme shown in Figure 3.21. Also, the concurrently designed control scheme had a 10% improvement in settling time and a steady-state error improvement of 12.5%.

Finally, Figure 3.24 compares the disturbance rejection capabilities of the sequentially and concurrently designed feedback controllers. Clearly, the sequentially designed controller is more capable of quickly rejecting disturbances. However, this is a result of the sequential design scheme over-meeting its time constant constraint in order to meet its overshoot constraint. The concurrent design scheme does not have this problem, and is able to just meet the time constant constraint and prioritize settling time.

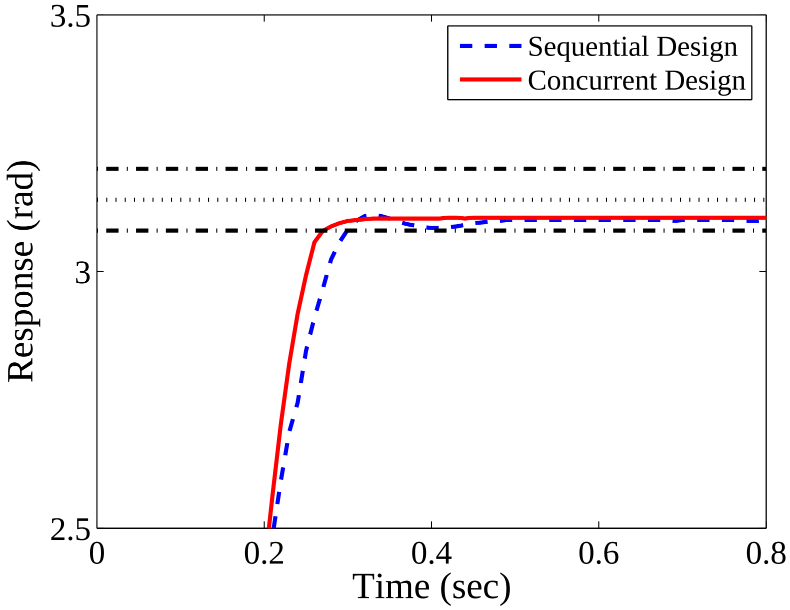


Figure 3.23: Step Response Comparison #2.

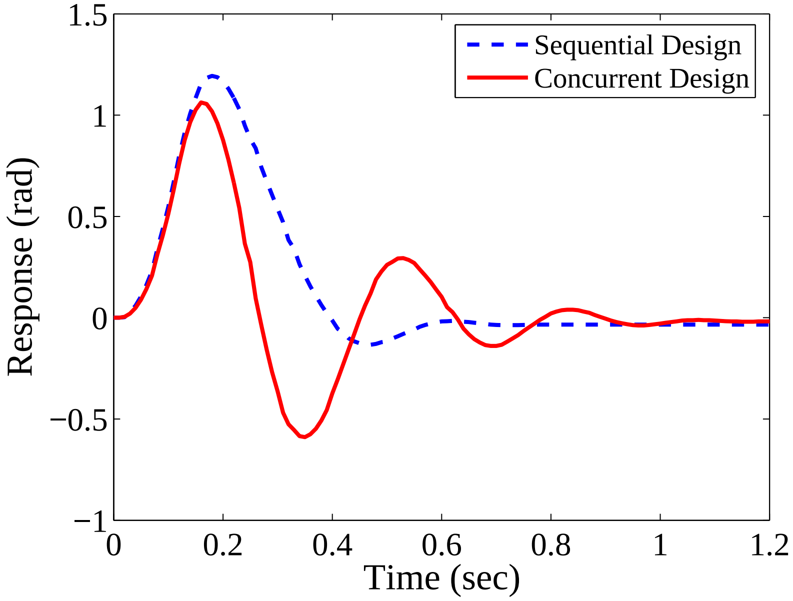


Figure 3.24: Disturbance Response Comparison.

3.2 Translational Mass Experiments

The concurrent design scheme for a mass plant was also tested experimentally on a portable crane setup located at the Georgia Institute of Technology and seen in Figure 3.25. Only one degree of freedom (the trolley motion) was utilized for these

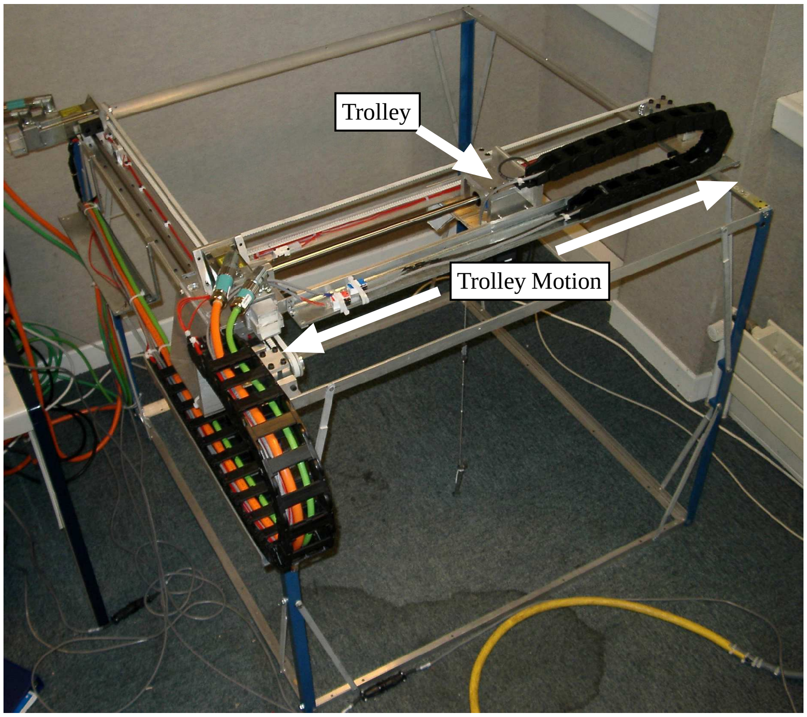


Figure 3.25: Portable Bridge Crane Located at Georgia Tech.

experiments. A rotational motor powered a belt system which connects to the trolley device. By way of this connection to the belt, the trolley translates back and forth as shown in Figure 3.25. As for the rotational mass experiments, the main nonlinearity for this system was coulomb friction, which here arose from the trolley's sliding contact with some of the crane's rigid, structural components. There was also a small, additional, oscillatory mode arising from the belt flexibility. Although the motor to trolley unit technically formed a two-mode, nonlinear system, it was approximated here as a translational mass.

3.2.1 Translational Mass: Experiment Set #1

Similar to Section 3.1.1, the mass of the trolley was estimated by fitting simulated responses to experimentally obtained responses. The response fitting program chose the proportional (K_p) gain, the derivative (K_d) gain and the mass which resulted in the best numeric fit. The results are presented in Table 3.4. The actual gain values

Table 3.4: Translational Mass System Verification

Actual K_p	Calculated K_p	Actual K_d	Calculated K_d	Calculated Mass (kg)
800	782	0	10.87	3.91
800	786	4	14.75	3.85
800	786	8	18.79	3.73
800	788	12	22.7	3.67
300	293	-6	6.2	3.82
300	294	-4	9.1	3.72
300	292	-2	11.4	3.94
300	292	0	13.23	3.79

are those which were used by the PLC to maneuver the trolley experimentally. The calculated values are those determined by the numeric fit program. As can be seen in the table, the proportional gain did not deviate significantly from the value used in the PLC. However, there was always a significantly higher calculated K_d value, due to the unmodeled friction. This discrepancy was taken into account when programming the PLC. That is, if a design scheme required a certain derivative gain, the PLC was given a smaller K_d value so that the total derivative gain (from the PD controller in the PLC and from friction) yielded the correct derivative action. Finally, it should be noted that the calculated mass values were fairly consistent, and the average mass value of $3.8kg$ was used for the sequential and concurrent design procedures.

Using the system verification results presented in Table 3.4, both a sequential and a concurrent design procedure were performed. The design constraints were an actuator limit of $30N$ (from the Siemens motors and drives used for these experiments), a maximum overshoot of 30% and a time constant upper limit of $0.25sec$. Within these constraints, the trolley was expected to move $0.1m$ with the fastest possible 2% settling time. Following these performance specifications and design constraints, the sequential and concurrent design solutions are presented in Table 3.5. Note that to account for the effects of friction, the K_d value was entered into the PLC as $K_d = 14.44$ for the sequentially designed controller and $K_d = 19.75$ for the concurrently designed controller. This was done via a process similar to the one whose results are shown in

Table 3.5: Translational Mass Design #1

	Concurrent Design	Sequential Design
K_p	830	300
K_d	30.4	30.4
ω_{cl}	15.3	8.88
ζ_{cl}	0.26	0.45
A_1	0.36	0.83
A_2	0.43	0.17
t_2	0.128	0.4
A_3	0.21	
t_3	0.313	

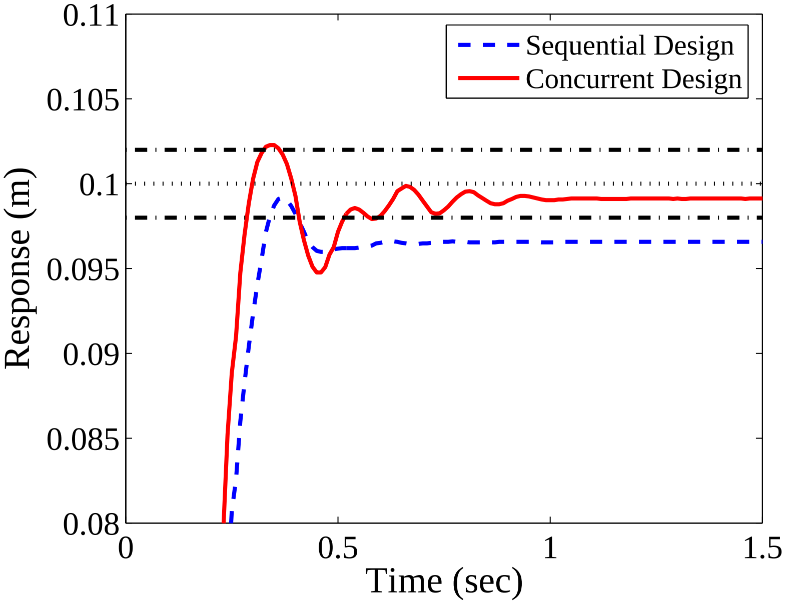


Figure 3.26: Step Response Comparison.

Figure 3.13, where the actual and simulated step responses were compared and the gains used in the PLC were adjusted so that the closed-loop system behaved as was designed by the sequential or concurrent design procedure. Also, as in Chapter 2, the sequential design scheme had an added t_r constraint to give the sequential design scheme the best possible performance.

These two controllers (sequential and concurrent) produce the step responses shown in Figure 3.26. Notice that while the concurrently designed controller is still somewhat oscillatory, the sequentially designed controller never settles within the 2%

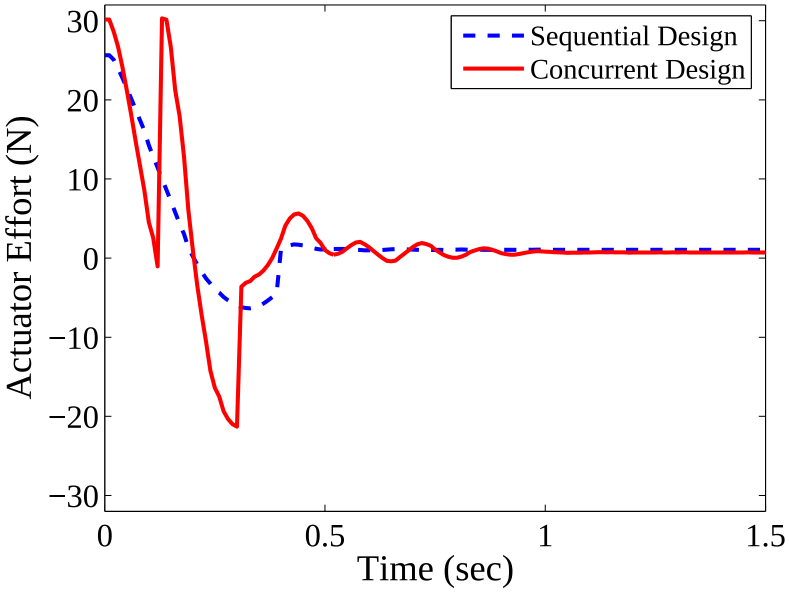


Figure 3.27: Actuator Effort Comparison.

settling time boundary, therefore yielding an infinite settling time. By using such a high proportional gain, the concurrently designed controller settles close to the desired location in a reasonable amount of time.

As described before, the cost required for such an improved response by the concurrently designed controller is a more aggressive actuator effort profile. The actuator efforts required for the previously shown step responses are shown in Figure 3.27. Neither controller violates the actuator limits, but the concurrently designed controller's actuator demands are more aggressive.

Lastly, Figure 3.28 shows that both control schemes are well equipped to reject disturbances in a short amount of time.

3.2.2 Translational Mass: Experiment Set #2

The second set of translating-mass experiments was performed using a slightly different trolley mechanism. Similar to the previously described experiments, this trolley's mass was estimated to be $3.46kg$. For this experiment, the system parameters and design constraints were:

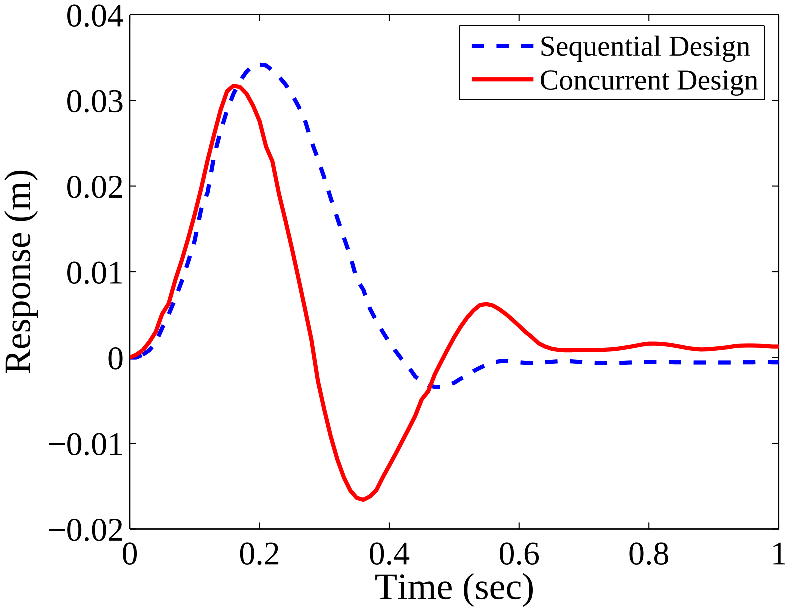


Figure 3.28: Disturbance Response Comparison.

System Parameters and Design Constraints

Step Size (L) = 0.1 m

Plant Mass (m) = 3.46 kg

Maximum Actuator Effort (U_{max}) = 30.25 N

Maximum Percent Overshoot (M_p) = 5%

Disturbance Rejection Constraint $(\frac{1}{\omega_{cl}\zeta_{cl}}) < 0.25$

Note that the actuator limit was set by the motor drives used for these experiments. The design results for this particular system are shown in Table 3.6. Again, the sequentially designed controller had a damping ratio too high to really benefit from input shaping.

Similar to the first translating-mass experiment, the derivative gain used in the PLC controlling the crane is not the effective K_d acting on the mass (due to friction). Therefore, before these controllers were tested, several more calibrations were

Table 3.6: Approximate Mass Experiments

	Concurrent Design	Sequential Design
K_p	843	302.4
K_d	27.7	44.65
ω_{cl}	15.6	9.35
ζ_{cl}	0.26	0.7
A_1	0.36	1
A_2	0.43	NA
A_3	0.21	NA
t_2	0.12	NA
t_3	0.3	NA

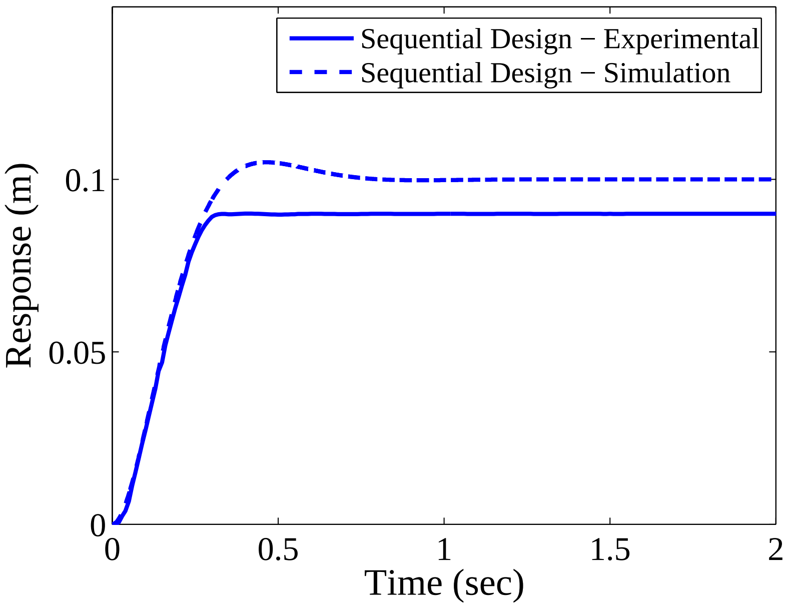


Figure 3.29: Sequential Design Experimental Results - Response.

completed to find the K_d value needed by the crane's PLC so that the trolley's motion would act like the closed-loop systems described in Table 3.6. The actual K_d values used in the crane's feedback control system were 35 (instead of 44.65) for the sequentially designed controller and 15 (instead of 27.7) for the concurrently designed controller.

Figure 3.29 shows the experimental response of the trolley under the sequentially designed controller. The actual response closely matches the desired response until stiction effects force it to stop and suffer from a significant steady-state error. This

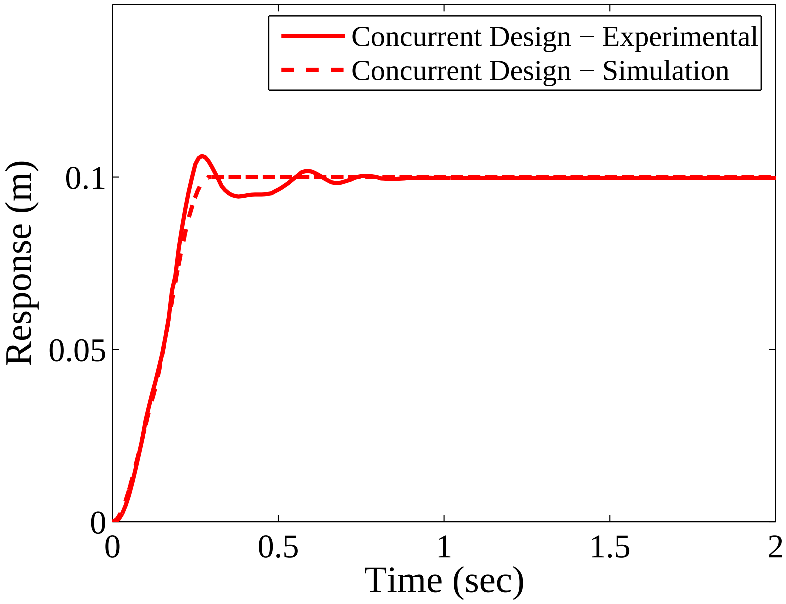


Figure 3.30: Concurrent Design Experimental Results - Response.

steady-state error problem could be improved by using friction compensating input shapers [40, 41]. However, this requires knowledge of the system's friction properties. This information is not always easy to determine. Fortunately, the concurrent design of OLIS and PID naturally mitigates the steady-state error problem by allowing for higher proportional gains.

Figure 3.30 shows the experimental response of the trolley under the concurrently designed controller. While this system moves fast and eliminates nearly all steady state error, there is some high-mode excitation. This results in a small overshoot and a longer settling time than expected.

Figures 3.31 and 3.32 show the intended and actual actuator efforts required for both the sequentially and concurrently designed controllers. Both figures show that the actual system behaves in a manner consistent with the theoretical predictions. The main difference for the sequential design is the non-zero, steady-state effort arising from friction. The main difference for the concurrent design is the high frequency component at the end of the move.

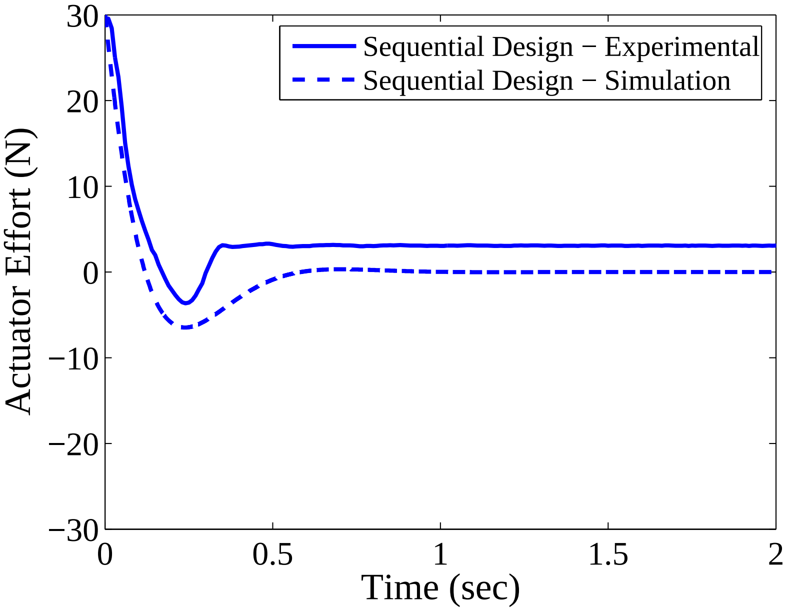


Figure 3.31: Sequential Design Experimental Results - Actuator Effort.

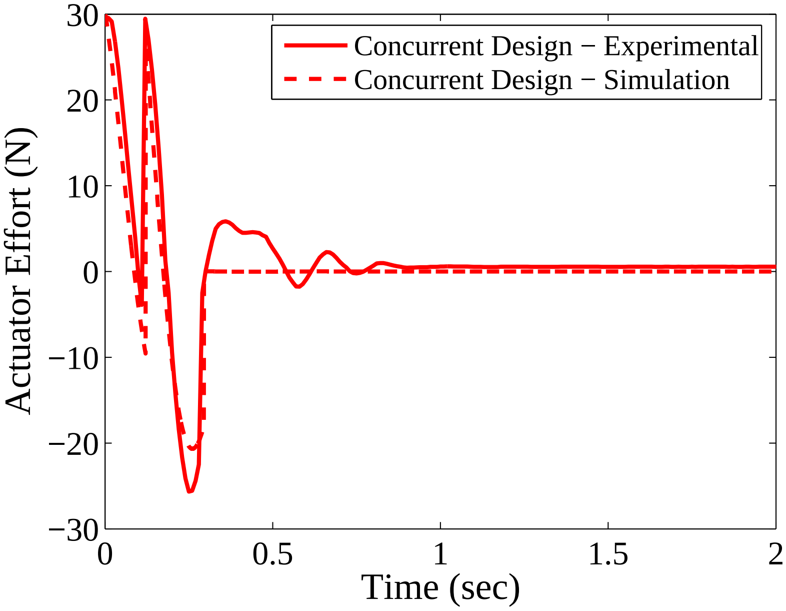


Figure 3.32: Concurrent Design Experimental Results - Actuator Effort.

Figure 3.33 directly compares the sequentially and concurrently designed responses. Clearly, the concurrently designed control scheme is faster than the sequentially designed scheme (in terms of rise time). Also, because of the higher K_p value typical of

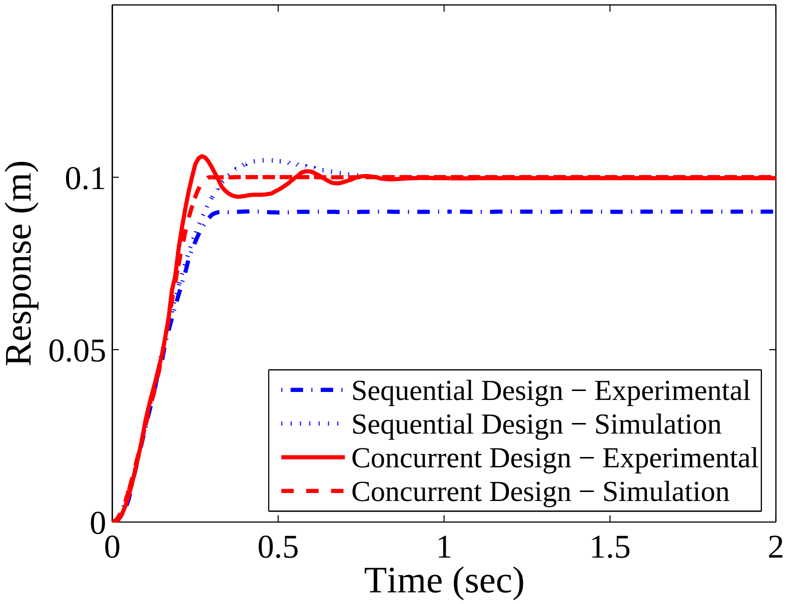


Figure 3.33: Step Response Comparison.

the concurrent design scheme, this controller is able to eliminate virtually all steady-state error. This was an unplanned benefit of using a concurrent design architecture that has occurred in each of the experiments presented in this chapter. As in the previous translational mass experiment, this steady-state error reduction allows the concurrent design scheme to have a finite 2% settling time of $0.53sec$, whereas the sequential design scheme never settles.

However, the concurrent design scheme's high K_p also causes some problems, including a small overshoot and settling time increase. Another cost associated with (and expected from) this particular concurrent design scheme is shown in Figure 3.34. The concurrently designed controller requires more actuator effort than the sequentially designed controller. However, neither controller saturates the actuator.

Figure 3.35 shows the disturbance rejection capabilities of the two control schemes. Clearly, both control schemes behave similar to the simulated predictions, with the sequentially designed controller yielding better results due to the fact that it exceeds its disturbance rejection constraint.

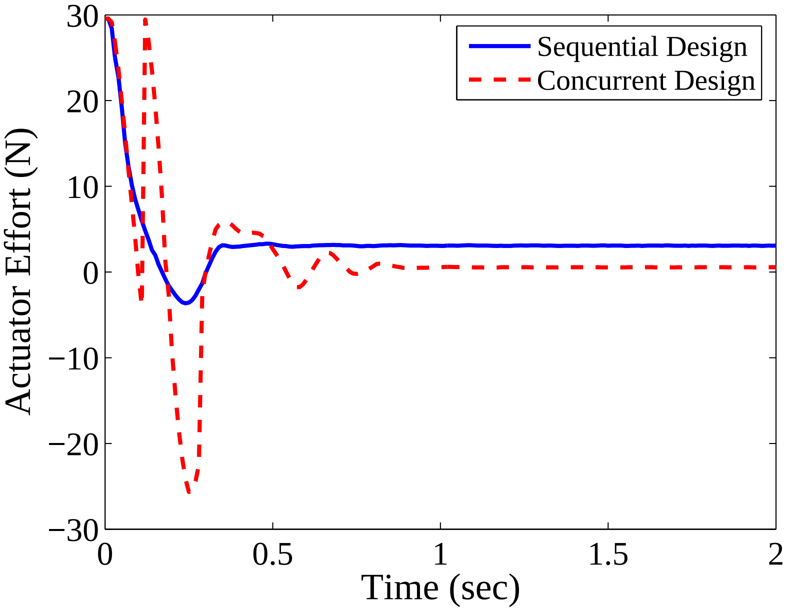


Figure 3.34: Actuator Effort Comparison.

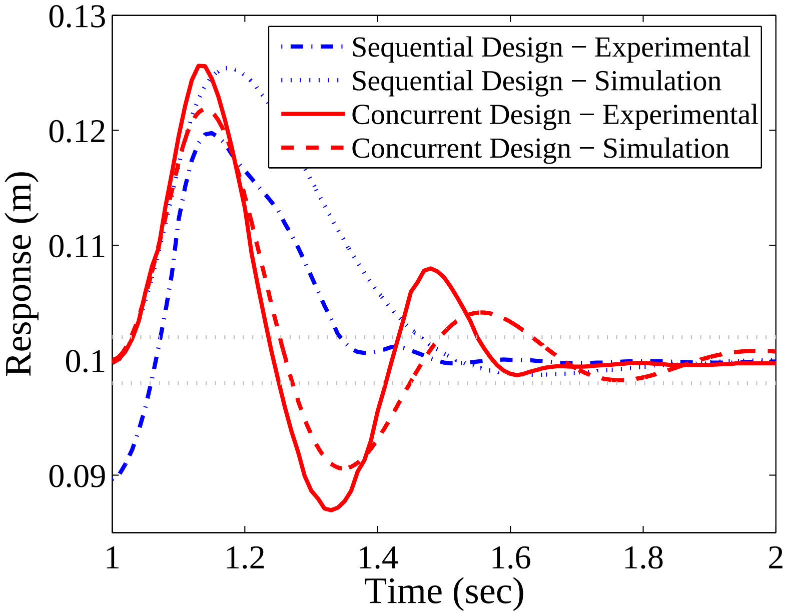


Figure 3.35: Disturbance Response Comparison.

There are several ways in which the un-modeled dynamics problem seen in the concurrently design controller may be addressed. One is to develop a more sophisticated plant model and concurrent design routine. This will be addressed in future

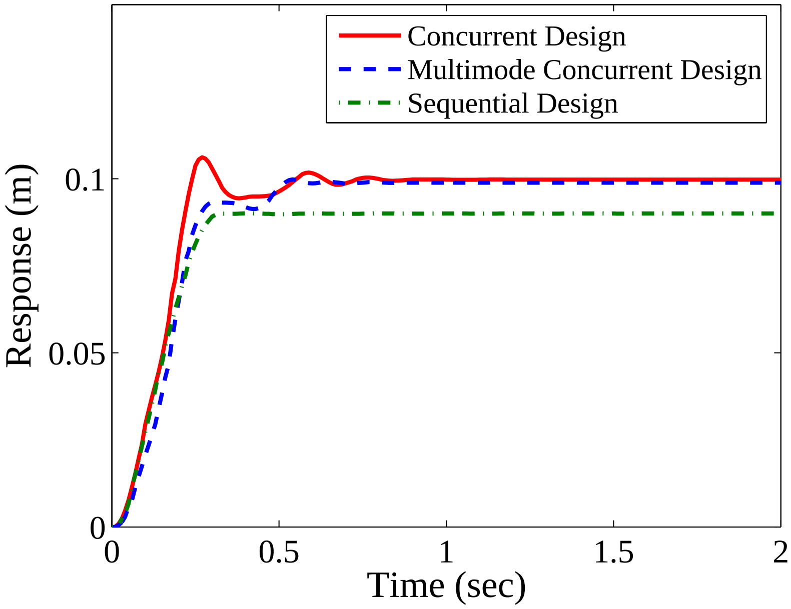


Figure 3.36: Multimode Shaper Experimental Results - Response.

chapters. Here, however, two simple approaches will be examined. The first approach simply analyzes the experimental response in Figure 3.30 and determines the frequency of the high-mode vibration. Then, a second, single-mode input shaper is added to the concurrently designed shaper for the purpose of eliminating this high mode. The response of this multi-mode concurrent design scheme is compared to the original concurrently and sequentially designed controllers in Figure 3.36. The second method simply re-ran the concurrent design routine and limited the K_p value. The idea was to compromise and create a system with a K_p value higher than the sequentially designed proportional gain (for a faster response) but lower than the original, concurrently designed proportional gain (to lesson the high-mode excitation). The resulting "K_p Limited" concurrent design scheme is compared to the sequentially designed controller and the original, concurrently designed controller in Figure 3.37. Both of these new responses eliminate the overshoot problem associated with the original concurrent design scheme and still improve upon the sequential design scheme's steady-state error problem. However, both new concurrent schemes still have settling times that are larger than expected.

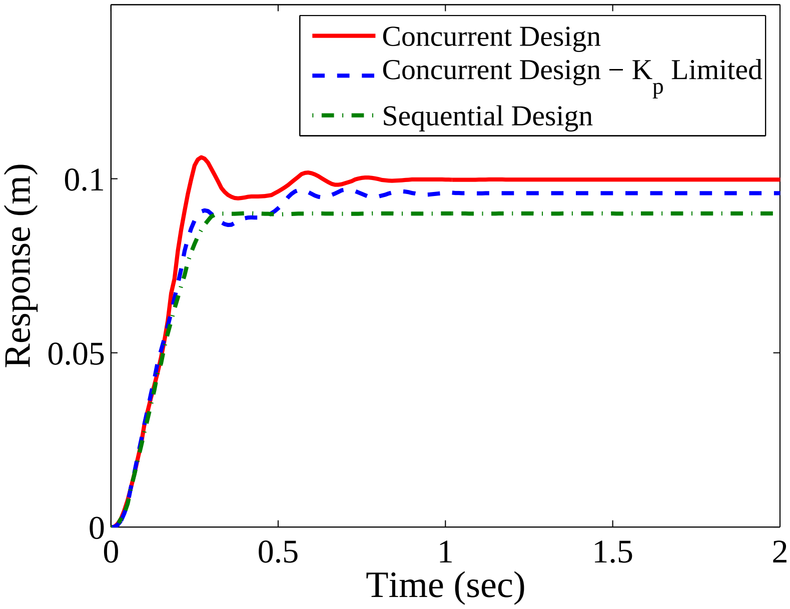


Figure 3.37: K_p Limited Experimental Response Comparison.

Even though some of the concurrent design results in this experiment deviated from theory, the results presented for this experiment are actually quite promising. It should be noted that a fairly complex, dynamic system, the crane trolley, was modeled with a very simple approximation (it was treated as a mass). Even with the gross simplification, the concurrent design procedure produced controllers that performed very well. The steady-state error was virtually eliminated and the overshoot was very small (or zero). And, these results were obtained while observing actuator limitations.

CHAPTER IV

CONCURRENT DESIGN - PURE AND APPROXIMATE SECOND-ORDER PLANTS

This chapter will continue the investigation of concurrently designing outside-the-loop input shaping and PD feedback controllers by looking at more complicated plants. As in the previous two chapters, the basic control architecture is as shown in Figure 4.1. Integral control is ignored again here so as not to add a first order term to the single-mode systems being studied. The only change in the block diagram is the "$\frac{1}{DCGain}$" term. This is done because some of the closed-loop systems studied in this chapter will not have unity DC gain when under PD control. Remember that the two previous chapters only studied mass plants, where the resulting closed-loop system under PD control always has a unity DC gain.

Section 4.1 will begin this chapter with a short extension of the concurrent design process to handle plants that are second-order oscillators. Section 4.2 will continue the extension of the concurrent design process to multi-mode systems with numerator dynamics that have the unique property of being reasonably approximated as single-mode plants without numerator dynamics. This section will also highlight

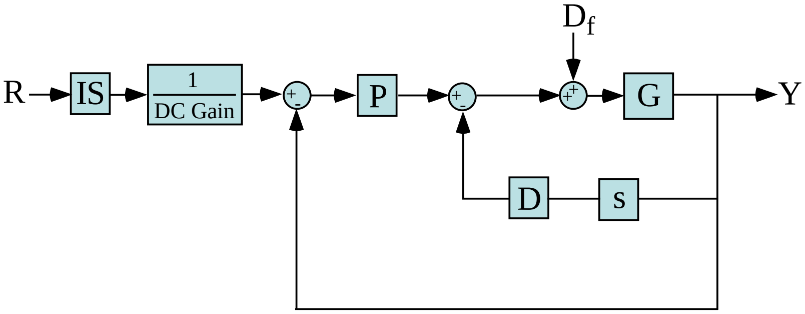


Figure 4.1: Proportional + Velocity Feedback Controller with Gain Compensator.

several scenarios described in Section 2.2.4; i.e. when concurrent and/or sequential design fails. Section 4.3 will then show how and when the single-mode approximation becomes invalid, motivating the work in the following chapter on non-reducible multi-mode systems.

4.1 Pure Second-Order Plants

This section will focus on plant types of the form:

$$G = \frac{A}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.1}$$

where ω_n is the plant natural frequency, ζ is the plant's damping ratio, and A is a scalar.

The mass studied in the previous chapter is very similar to the second-order plant studied here. In fact, a mass is just a non-oscillatory second-order system. When a mass is included within a PD feedback loop, the closed-loop system is second-order and oscillatory, just like the plant given in (4.1) (assuming the closed-loop system remains under-damped). Therefore, it makes sense that the equations of motion for the closed-loop system with G as a second-order oscillator are similar to the equations for the closed-loop system when G is a mass. However, there is one important difference. The equation describing the actuator effort becomes more complicated when the plant is a second-order oscillator. The maximum actuator effort is no longer guaranteed to occur at time $t = 0$. This complicates the equations and design procedure, which must check for actuator saturation. While there are many ways to address this problem, a full search routine is utilized here.

This full search routine was chosen here so as to highlight the differences between an analytically based solution routine and a comprehensive, numerical search routine. The analytical solution (as discussed while studying the mass plant in the previous two chapters) is very fast. The numerical search routine utilized here is much slower, but is better able to handle more complicated systems not easily described analytically. Studying these advantages and disadvantages will lead to a future, compromise solution that relies heavily on equations to describe a closed-loop system's response (its overshoot, maximum actuator effort, etc.). This compromise solution will then periodically check the analytically based solution via a full, numeric simulation of the system.

The search routine utilized in this section establishes K_p and K_d ranges and then performs an exhaustive search throughout this parameter space for the optimal K_p and K_d values. The minimum K_p is chosen from a Routh stability formulation. The minimum K_d is chosen from both a Routh stability formulation and the minimum time-constant constraint (for disturbance rejection). Note that the K_p minimum is sometimes negative. The maximum K_p is chosen to be the maximum proportional gain which does not cause actuator saturation at time $t = 0$. While this time is no longer the time at which U_{max} necessarily occurs, it still yields an obvious upper limit on K_p. Finally, the maximum K_d value is the derivative gain that yields a critically damped system when K_p is held at its maximum possible value. The closed-loop transfer function is forced to remain oscillatory because this is the region where concurrent design is superior to sequential design. It has already been discussed in Section 2.2.4 that if the design constraints force the closed-loop system to be overdamped, concurrent and sequential designs would yield equal solutions.

Once the K_p and K_d boundaries have been established, the search routine tests every K_p/K_d combination via a numeric simulation. Obviously, the sequential design technique only simulates the step response of the PD controlled plant, because the input shaper (a standard ZV shaper in this section) is not determined until after the PD gains are chosen. The concurrent design, however, utilizes the newly developed shaper presented in Section 2.3.5 and simulates the response of the full OLIS/PD controller. The simulations numerically (not analytically via equations) determine overshoot, maximum actuator effort and settling time. Again, this design routine attempts to meet disturbance rejection, overshoot, and maximum actuator effort constraints while minimizing settling time.

As an example, an OLIS/PD controller design was performed with the system parameters and constraints are set to:

Step Size (L) $= 1$

Plant Frequency (ω_n) $= 3\ \frac{rad}{s}$

Plant Damping (ζ) $= 0.1$

Plant Numerator (A) $= 2$

Maximum Actuator Effort (U_{max}) $= 100$

Maximum Percent Overshoot (M_p) $= 10\%$

Disturbance Rejection Constraint ($\frac{1}{\omega_{cl}\zeta_{cl}}$) < 0.2

Note that ω_{cl} and ζ_{cl} refer to the system's closed-loop damping and natural frequency. The resulting controller parameters are shown in Table 4.1. Note that two sequential designs were performed here. The first gives no importance to rise time and uses the feedback controller to minimize the 2% settling time. The second sequential design

Table 4.1: Design Results: Second-Order Oscillator

	Concurrent Design	Sequential Design #1	Sequential Design #2
K_p	132.4	90.5	90.5
K_d	4.7	10.55	8.3
ω_{cl}	16.54	13.78	13.78
ζ_{cl}	0.3	0.79	0.62
A_1	0.73	1	0.92
A_2	0.14	NA	0.08
A_3	0.13	NA	NA
t_2	0.1988	NA	0.29
t_3	0.1996	NA	NA

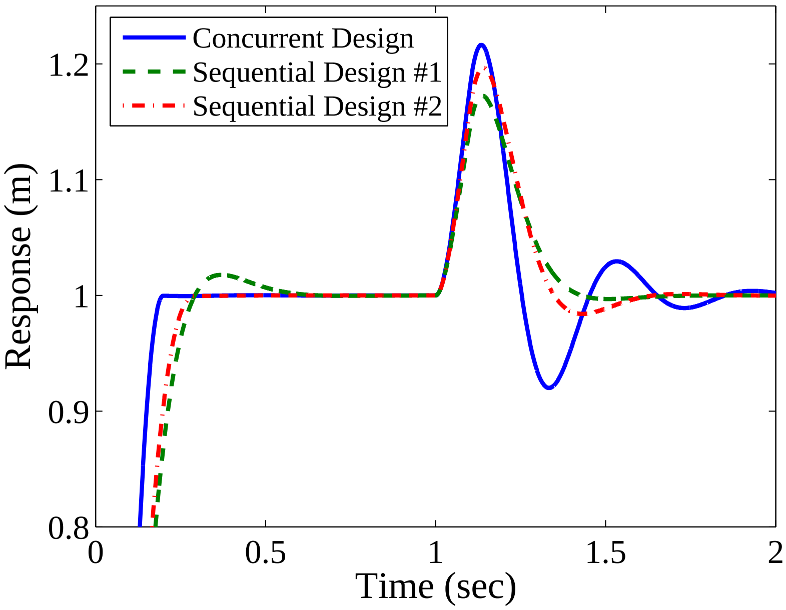


Figure 4.2: Design Comparison for a Second-Order Plant - Step Response.

minimizes settling time but enforces a minimum rise time constraint. As in previous sections, this is done to produce a more competitive sequential design scheme for comparison to the newly proposed concurrent design strategy. One obvious difference between concurrent and sequential designs can be seen in Table 4.1. The concurrent design scheme typically yields higher K_p and lower K_d values than the sequential design schemes. This results in a concurrently designed closed-loop system with higher natural frequency and lower damping. Note that the first sequential design scheme produced a closed-loop system whose damping was high enough to negate the need for an input shaper - the resulting system is simply not oscillatory enough.

The performance of all three controllers is depicted in Figures 4.2 and 4.3. Figure 4.2 shows the step response and disturbance rejection capabilities of each controller. Note that the disturbance force was a quick pulse acting on the plant, as described by the D_f signal in Figure 4.1. Figure 4.2 shows that the concurrent design is 33% faster, in terms of settling time, than “Sequential Design #1” and 28% faster than “Sequential Design #2”. However, the sequential designs have a better disturbance

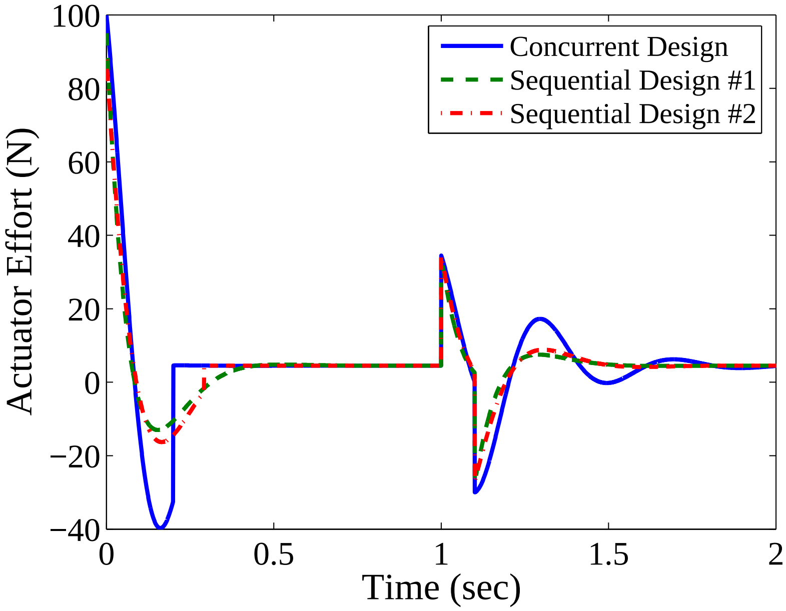


Figure 4.3: Design Comparison for a Second-Order Plant - Actuator Effort.

rejection capability. As in the design for a mass plant, this is a result of the sequential designs exceeding their disturbance rejection requirements in an attempt to meet their overshoot requirements by utilizing only the PD, feedback controller. The concurrent design scheme uses both the feedback controller and the input shaper to lessen overshoot. Therefore, it does not need to exceed its disturbance rejection constraint. This allows the concurrent design scheme to focus on minimizing settling time. The required actuator efforts for these responses can be seen in Figure 4.3. Again, the cost of moving faster is an increase in actuator effort.

4.2 Fourth-Order Plant with Numerator Dynamics

Many real world systems (nonlinear and multi-mode) are reasonably approximated by their dominant, oscillatory mode. This section will discuss a concurrent design strategy specifically for these kinds of systems. The design routine will determine the dominant mode and use an analytic approach to satisfy design constraints. Once a solution has been found, a numerical simulation will check the full (multi-mode) system response to verify that the design constraints have been met. If the unmodeled dynamics have caused the system to violate any constraints, the analytical solution routine will be re-run under increasingly stringent design constraints until a satisfactory solution has been obtained.

This type of design routine is a comprise between the two architectures discussed in this, and previous, chapters. The concurrent design routine for the mass was purely analytical. As such, it quickly produced a solution. The design scheme for the second-order plant was purely numerical, simulating every K_p/K_d combination and verifying constraints. While this routine was very thorough, it takes significantly longer than the analytical approach. The routine presented here is a compromise, in that it uses purely analytical criteria to produce PD and input shaper parameters. Then, it tests this design numerically, and re-runs the analytical routine under more stringent design constraints if the numeric check fails.

It should also be noted that this section utilizes robust input shapers: the ZVD and the EI shapers. Because the system is actually multi-mode and has numerator dynamics, it is desirable to use robust shapers that can handle some of the effects of unmodeled dynamics.

4.2.1 Description of Fourth-Order Plant with Numerator Dynamics

The plant studied in this section has a transfer function of the form:

$$\frac{\omega_1^2(s+a)}{(s^2+2\zeta_1\omega_1 s+\omega_1^2)(s^2+2\zeta_2\omega_2 s+\omega_2^2)} \tag{4.2}$$

This is a fourth-order system with two oscillatory modes and one numerator zero. This plant is chosen because it is clearly not a simple, single-mode system like the ones studied thus far. However, when placed inside a PD feedback loop, the closed-loop system will be approximated as a single oscillatory mode with no numerator dynamics. This is justified for several reasons. First, the derivative control is again implemented via velocity feedback. Therefore, the PD controller does not introduce its own closed-loop numerator. Secondly, the plant zero and mode represented by ω_2 and ζ_2 are located much farther from the real-imaginary plane origin than the single mode represented by ω_1 and ζ_1. Thirdly, "a" is large, meaning the closed-loop zero at $s = -a$ will not significantly affect the closed-loop system's response. Note that the zero is also assumed to be in the left-half plane, creating a minimum-phase plant. Because the resulting closed-loop system is approximated as a second-order system, the PD gains and input shaper parameters are chosen analytically via the well-known equations for time domain characteristics of second-order systems.

Comparing concurrent to sequential design with this plant type, it is easy to demonstrate three important levels of concurrent design success/superiority. Case 1 will demonstrate an example when concurrent design yields a controller that is different from, and superior to, a sequentially designed controller. Case 2 will show an example of a constraint set that causes the concurrent design to yield a solution equal to a sequential design solution. Finally, Case 3 will show an example when sequential design is unable to obtain a viable solution, but concurrent design is successful in meeting all constraints.

It is important to note that because the single-mode concurrent design scheme utilized here is based upon an approximation and simplification of the actual system, it will not always yield the optimal solution. A much slower, fully numeric routine (full-search concurrent design) is also utilized here to find the optimal, concurrent solution and compare that answer with the near-optimal, single-mode concurrent solution based upon the simple approximation. These two concurrent design schemes present a typical tradeoff between solution accuracy and difficulty.

4.2.2 Case 1 - Concurrent Design Superior to Sequential Design

The system parameters and design constraints for this test are:

Step Size (L) = 1

$\omega_1 = 2\pi \ \frac{rad}{s}$

$\zeta_1 = 0.01$

$\omega_2 = 30\pi \ \frac{rad}{s}$

$\zeta_2 = 0.07$

a = 100

Maximum Actuator Effort (U_{max}) = 300

Maximum Percent Overshoot (M_p) = 20%

Disturbance Rejection Constraint ($\frac{4}{\omega_{cl}\zeta_{cl}}$) < 50 sec

The concurrent and sequential design results are shown in Table 4.2. As expected, both concurrent design routines yield closed-loop systems that are much more oscillatory than the sequentially designed controller. In fact, because the sequential design scheme yielded such a high damping ratio, the second step in the design process utilized no input shaper. Also as expected, the full-search concurrent design scheme yielded a result similar to, although slightly different from, the single-mode concurrent design scheme. This occurs because of the full-search's recognition of the second

Table 4.2: Case 1 Design Results

	Full-Search Concurrent Design	Single-Mode Concurrent Design	Sequential Design
K_p	591	591	51
K_d	0	10	30
ω_{cl}	17.7	17.29	7.4
ζ_{cl}	0.07	0.2	0.82
A_1	0.31	0.43	1
A_2	0.49	0.45	NA
A_3	0.20	0.12	NA
t_2	0.18	0.19	NA
t_3	0.36	0.37	NA

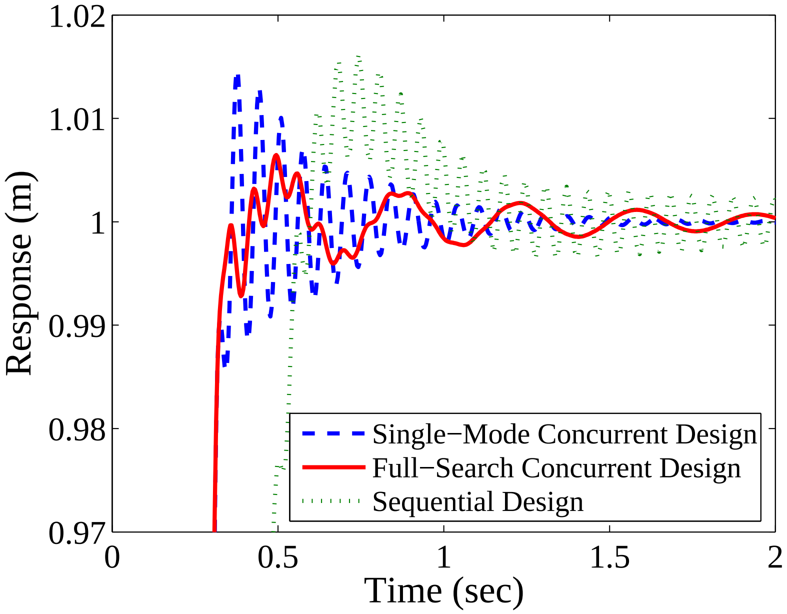


Figure 4.4: Case 1 Step Response Comparison.

mode and numerator dynamics. Fully simulating the complete system will often provide a more-optimal solution than the approximation based, single-mode concurrent design scheme.

Figure 4.4 compares the step responses for the concurrently and sequentially designed controllers. In terms of the 2% settling time criteria, the concurrent design schemes are approximately 40% faster than the sequential design scheme. The full-search concurrent design scheme is marginally faster than the single-mode concurrent design scheme and has less overshoot.

The cost for these faster responses is shown in Figure 4.5. While both concurrent design schemes stay within the actuator effort boundaries, the actuator demands are certainly higher for the concurrently designed controllers.

Interestingly, Figure 4.6 shows that the single-mode concurrent design technique has better disturbance rejection than the full-search concurrent design technique. However, all three designs are well within the 50sec disturbance rejection constraint.

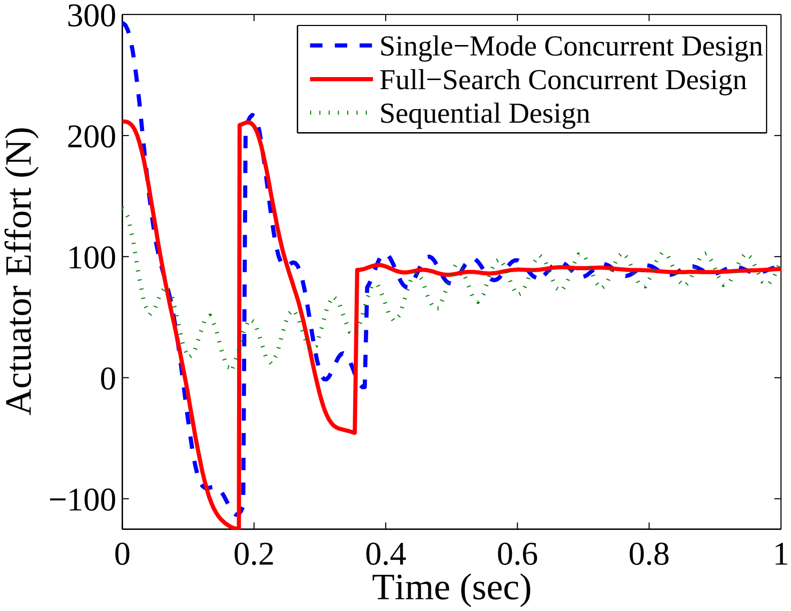


Figure 4.5: Case 1 Actuator Effort Comparison.

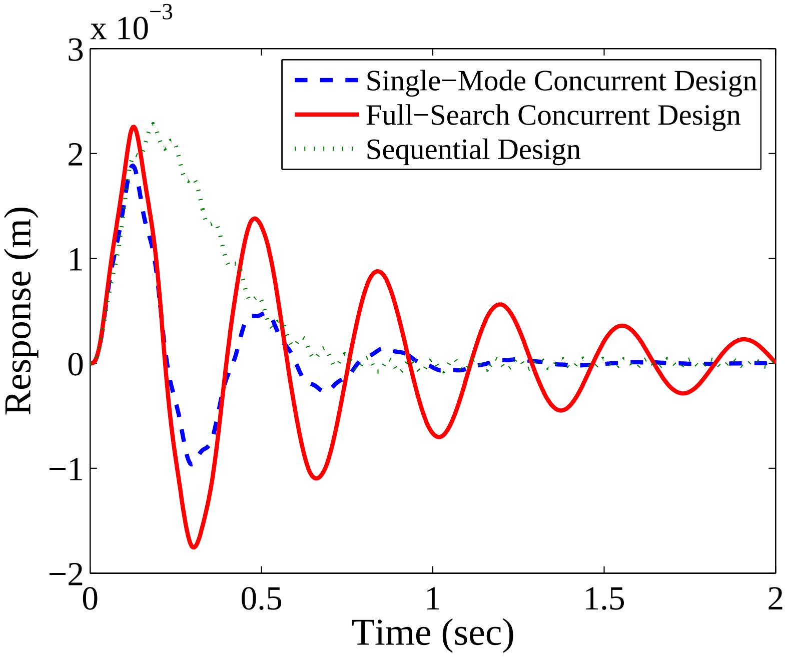


Figure 4.6: Case 1 Disturbance Response Comparison.

This particular concurrent design scheme also utilizes robust input shapers (ZVD shapers in Case 1) that can handle reasonable errors in the estimation of system natural frequencies. Figure 4.7 shows the concurrently designed step response when

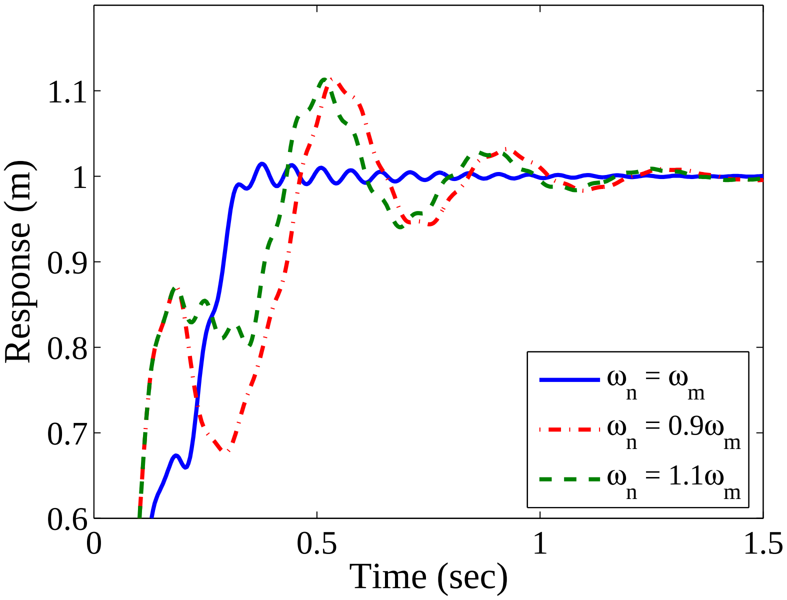


Figure 4.7: Case 1 Modeling Error Response Comparison.

the input shaper is designed with a 10% error in its estimation of the closed-loop system's dominant natural frequency. Because of the robustness properties of the input shaper, even a ± 10% error results in a step response well within the 20% overshoot constraint.

4.2.3 Case 2 - Concurrent Design Equals Sequential Design

In Case 2, the design constraints force the closed-loop system to be heavily damped by greatly decreasing the allowable time for a disturbance to be rejected. Remember that disturbance rejection is solely a property of the closed-loop system, and is unaffected by the presence or absence of an input shaper outside of the loop. Therefore, in order to meet this new disturbance rejection constraint, each of the three design schemes were forced to have a heavily damped closed-loop system. This meant that input shaping was unwarranted, resulting in equal solutions by both of the concurrent design schemes and the sequential design technique.

The system parameters and design constraints for this test are:

Step Size (L) $= 1$

Table 4.3: Case 2 Design Results

K_p	171
K_d	60
ω_{cl}	8.5
ζ_{cl}	0.8

$\omega_1 = 2\pi\ \frac{rad}{s}$

$\zeta_1 = 0.01$

$\omega_2 = 40\pi\ \frac{rad}{s}$

$\zeta_2 = 0.1$

a = 100

Maximum Actuator Effort (U_{max}) = 400

Maximum Percent Overshoot (M_p) = 6%

Disturbance Rejection Constraint ($\frac{4}{\omega_{cl}\zeta_{cl}}$) < 1 sec

The concurrent and sequential design results are shown in Table 4.3.

4.2.4 Case 3 - Concurrent Design Works, Sequential Design Fails

It is easy to imagine scenarios when PID control is incapable of meeting design constraints. For example, a PID controller could easily fail to produce a fast system under rigid actuator constraints. One of the interesting results from this research is that concurrent design expands PID control's realm of useful applications, because this design scheme combines the PID feedback controller with an input shaper. This should make sense considering that sequential design (in its first step) relies solely on the PID controller's abilities to meet performance constraints (it only then adds an input shaper in the next phase of the design procedure). On the other hand, concurrent design allows input shaping to help the PID controller and vice-versa. By using more “tools”, more can be accomplished. The following example illustrates this principle.

The system parameters and design constraints for this case are:

Step Size (L) = 1

$\omega_1 = 2\pi \ \frac{rad}{s}$

$\zeta_1 = 0.01$

$\omega_2 = 14\pi \ \frac{rad}{s}$

$\zeta_2 = 0.005$

a = 100

Maximum Actuator Effort (U_{max}) = 400

Maximum Percent Overshoot (M_p) = 6%

Disturbance Rejection Constraint ($\frac{4}{\omega_{cl}\zeta_{cl}}$) < 50 sec

The single-mode concurrent design results are shown in Table 4.4.

Note that the sequential design solution is not shown because the sequential design scheme was unable to determine a solution. The plant parameters and performance

Table 4.4: Case 2 Design Results – Single-Mode Concurrent Design

K_p	11
K_d	0
ω_{cl}	7.9
ζ_{cl}	0.02
M_p	0.048
A_1	0.28
A_2	0.47
t_2	0.4
A_3	0.25
t_3	0.8

constraints created a scenario where PD control was insufficient to create an acceptable solution. The reason was that the feedback controller was not able to produce gains high enough to meet the overshoot constraint while maintaining stability. Since input shaping can significantly lower overshoot, both concurrent design schemes were not forced to meet this overshoot constraint via the feedback controller alone. Subsequently, they found an acceptable solution. Also note that the full-search concurrent design scheme yielded the same solution as the single-mode concurrent design scheme.

Recall from Section 2.2.3 that this research sought to use input shapers with one independent variable. This scenario is an excellent example of the use of such an input shaper. The input shaper used in this analysis was an *EI* shaper. As mentioned in Chapter 1, *EI* shapers can be designed to have any insensitivity value - which can either be directly specified or indirectly specified by setting the *EI* shaper's maximum percent residual vibration. This concurrent design scheme adjusted the *EI* shaper's maximum percent residual vibration to ensure that the overall system's overshoot constraint was met while maintaining the highest possible insensitivity. While the closed-loop system was allowed up to 6% overshoot, the concurrent design scheme yielded an *EI* shaper that allowed only 4.8% vibration. This resulted from the concurrent design scheme's numeric check which included the small amount of additional overshoot added by the second mode. Because the second mode is not completely negligible, the concurrent design scheme scaled down the overshoot limit, iterating the design with smaller and smaller amounts of vibration allowed by the *EI* shaper until the numeric check of the full system confirmed an overshoot less than 6%.

Figure 4.8 shows the step response for the concurrently designed controller in Case 3. Note that because an *EI* shaper is used, some vibration will be present even when the dominant mode's frequency is perfectly known - not to mention there will always be some amount of vibration due to the unmodeled second mode. The advantage

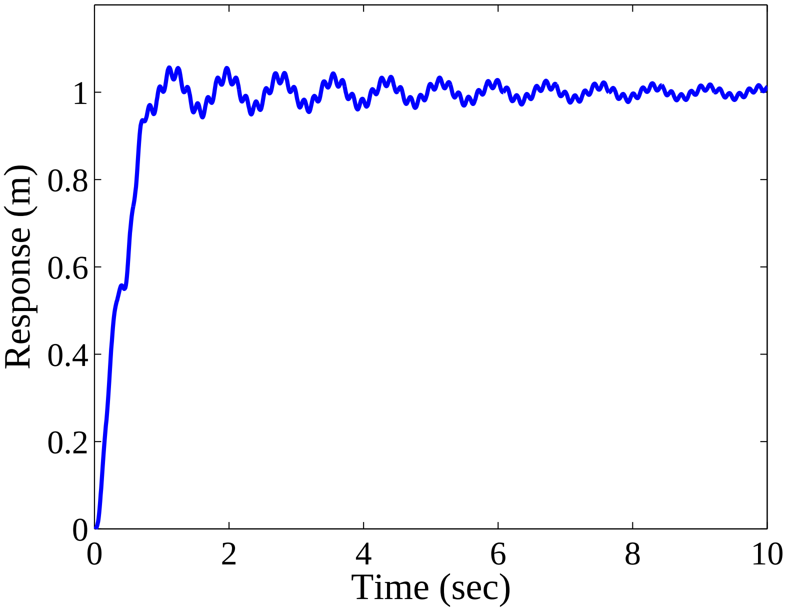


Figure 4.8: Case 3 Step Response.

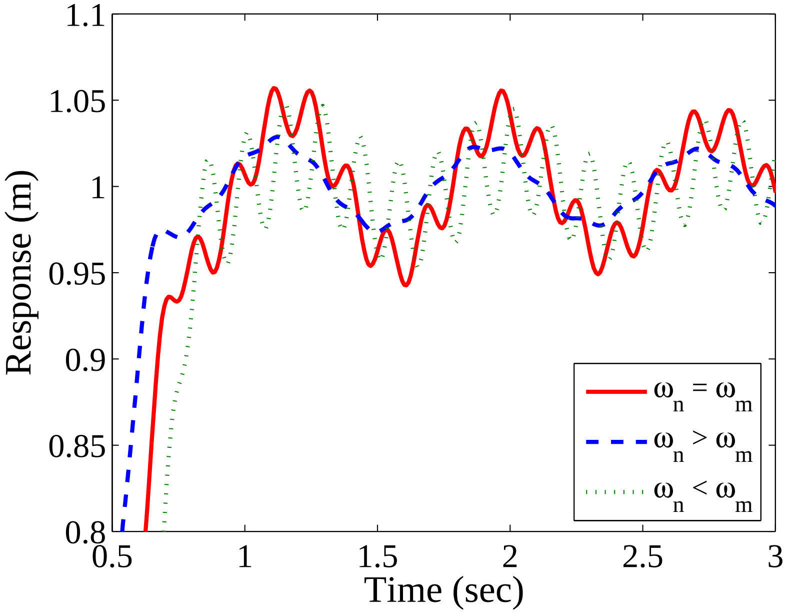


Figure 4.9: Case 3 Modeling Error Response Comparison.

gained from this is a large insensitivity to modeling errors. Figure 4.9 shows step responses under the same modeling error conditions that were investigated in Case 1. The difference is that the modeling errors in Case 1 resulted in more vibration than when the frequency was perfectly known. Here, by using an *EI* shaper, modeling

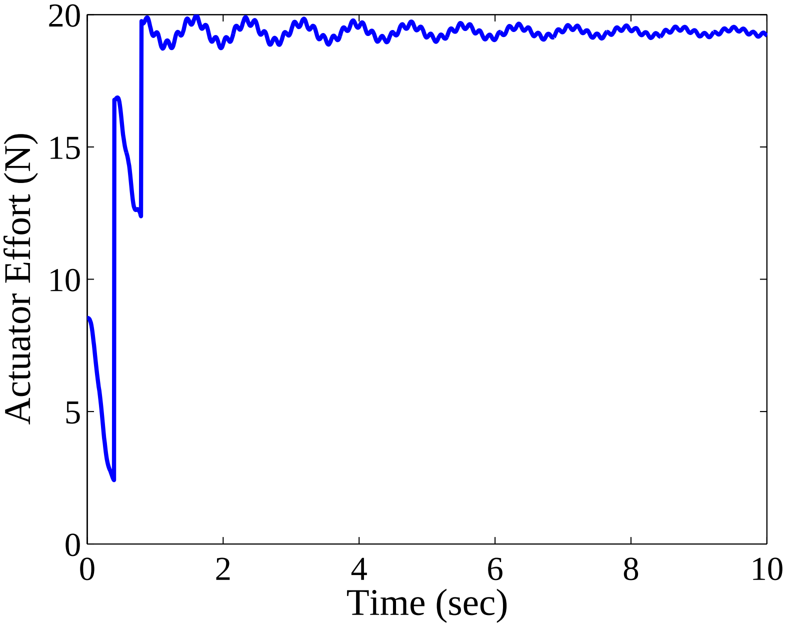


Figure 4.10: Case 3 Actuator Effort Comparison.

errors result in equal or lesser amounts of vibration than when the model is perfectly known. Of course, for extremely large errors, the vibration would actually be larger than when the dominant mode is perfectly known. By allowing an acceptable level of vibration at the modeled frequency, EI shapers have more insensitivity than ZVD shapers (like the one used in Case 1) and can often result in less than predicted vibration when implemented on a real machine due to moderate modeling errors.

Figure 4.10 shows that the concurrently designed controller stayed well within the actuator limits. Note that the actuator is vastly under-utilized because the particular closed-loop system used in this case required small gains to maintain stability. Finally, Figure 4.11 shows that the concurrently designed controllers are able to reject disturbances in the allowed amount of time.

4.3 When the Second-Order Approximation Fails

Unfortunately, the single-mode approximation used throughout this chapter is not always a reasonable assumption. Many systems have multiple, non-negligible modes

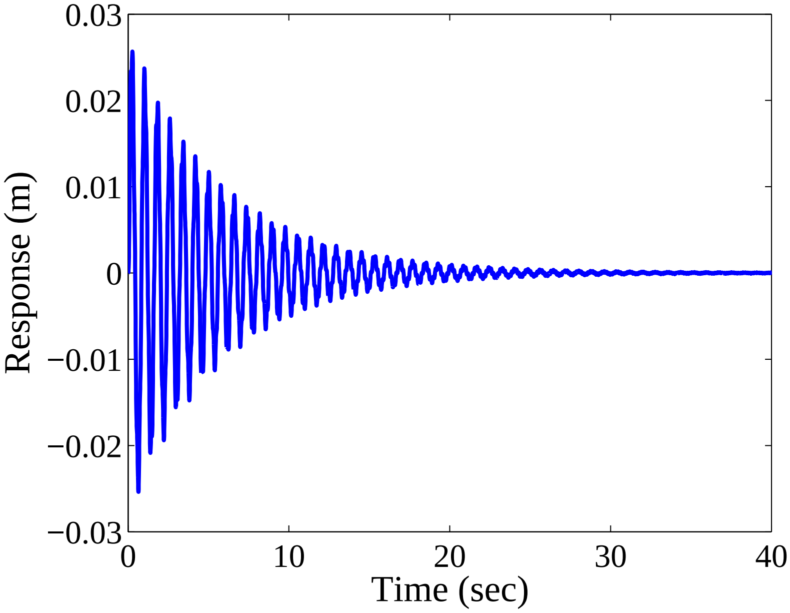


Figure 4.11: Case 3 Disturbance Response Comparison.

or non-negligible numerator dynamics. In these cases, the single-mode concurrent design scheme which assumes one mode and zero numerator dynamics is insufficient. This section will demonstrate when a secondary mode becomes significant and cannot be neglected in the concurrent design.

Revisiting the plant used in Section 4.2,

$$G = \frac{\omega_1^2(s+a)}{(s^2+2\zeta_1\omega_1 s+\omega_1^2)(s^2+2\zeta_2\omega_2 s+\omega_2^2)} \tag{4.3}$$

the mode resulting from ω_2 and the closed-loop zero resulting from the $s+a$ term were negligible compared to the dominant, low mode in the three previously considered cases. Here, the value of ω_2 will be slowly decreased until this second, flexible mode is no longer negligible. The plant parameters and design constraints will be as follows:

Step Size (L) $= 1$

$\omega_1 = 2\pi\ \frac{rad}{s}$

$\zeta_1 = 0.01$

$\omega_2 = \mathrm{R}\pi \ \frac{rad}{s}$

$\zeta_2 = 0.07$

a = 100

Maximum Actuator Effort (U_{max}) = 300

Maximum Percent Overshoot (M_p) = 8%

Disturbance Rejection Constraint ($\frac{4}{\omega_{cl}\zeta_{cl}}$) < 10 sec

Here, R is varied from 40 down to 20, eventually resulting in a non-negligible second mode.

There are three main OLIS/PID design techniques that will be tested here. Just as in the previous section, which analyzed the same plant type under a second-order approximation, ZVD and EI shapers will be used. The first design technique is the single-mode concurrent design technique that was described in Section 4.2. This design technique concurrently solves for the PD controller gains and input shaper parameters in light of the assumption that only one mode is significant. This routine uses the well known equations describing the motion of second-order systems as well as the known effects of input shaping. The second design technique is a single-mode sequential design procedure also described in Section 4.2. This procedure also uses the equations describing second-order systems, but does not include the effect of input shaping. However, the full, numeric checks that complimented the single-mode concurrent and sequential design techniques in Section 4.2 are eliminated here. This is done to force the design schemes to rely solely upon the single-mode approximations (no numeric simulation of the full, two-mode system is allowed).

The third design technique is the full-search concurrent design procedure that was also discussed in Section 4.2. This version of a concurrent design scheme requires significantly more time to execute, but is not limited by the single-mode assumption. This full-search concurrent design scheme simulates the full 4^{th}-order system via Simulink for each controller/shaper parameter that is free to vary. The simulated response is then analyzed for its performance characteristics. By discretizing the solution space and searching over all possible parameter combinations, the full-search concurrent design scheme yields the optimal solution over the solution space used. The two concurrent design schemes will be compared throughout this section to see how close the single-mode concurrent design scheme can come to the slower, but more robust, full-search scheme. They will also be compared to see when the single-mode concurrent design scheme fails (as the neglected mode becomes non-negligible). Finally, the computation time of each solution routine will be calculated. This will demonstrate the tradeoff between using a single-mode, approximation based routine and a complicated, but more accurate, routine. As one would expect, the cost of accuracy and confidence in the solution is computation time.

Figure 4.12 shows the settling times achieved by each of the two concurrent design

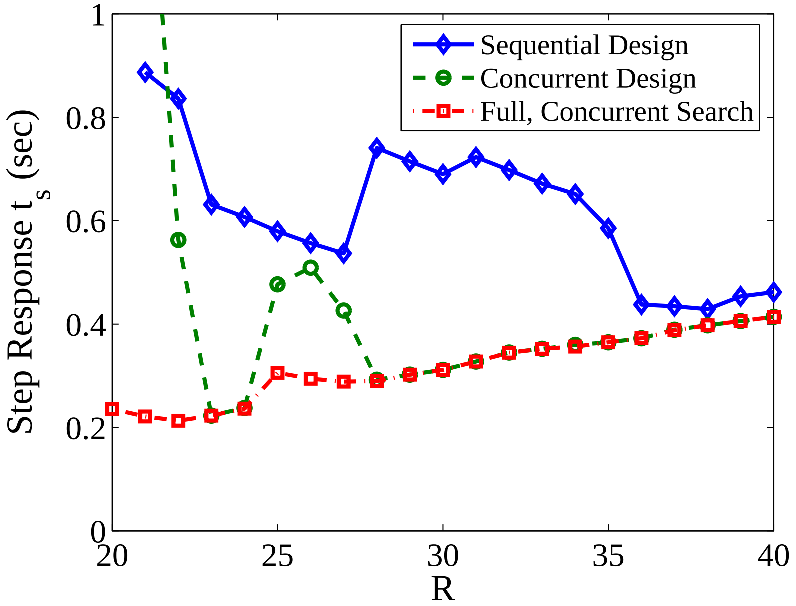


Figure 4.12: Settling Time Comparison.

schemes. Remember that for these designs, settling time was minimized within the boundaries established by other performance constraints (M_p, disturbance rejection and U_{max}). Also, for the sake of comparison, the settling times achieved by the sequential design technique are shown. Notice that when R is high (the second mode is almost negligible), the single-mode concurrent design technique (labeled as "Concurrent Design") yields results that are equal to the full-search concurrent design (labeled as "Full, Concurrent Search") and superior to the sequential design technique. Obviously, the single-mode approximation is valid for these high R values.

When the value of R becomes low (below 30 in this example), the single-mode approximation becomes inaccurate. At first, the single-mode concurrent design technique still yields reasonable settling times. However, these settling times are often inferior to those achieved by the full-search concurrent design technique. Eventually, as R approaches 20, the single-mode concurrent design technique yields control schemes with extremely high settling times. This is caused by the now significant second mode resulting in a significant amount of vibration. Obviously, the full-search concurrent design technique is able to avoid this problem because it always simulates the full, fourth-order system and takes the second mode into account. Interestingly, the sequential design technique, while never assuming a second mode exists, also does not suffer from the extremely high settling time problem associated with the single-mode concurrent design technique. This is caused by the conservative nature of the sequential design technique, which normally creates relatively high-damped closed-loop systems (because it must meet constraints and performance specifications solely with the feedback controller).

In addition to its failure to yield competitive settling time values when R is low, the single-mode concurrent design scheme under low R values begins to violate its constraints. Figure 4.13 shows the overshoot of each of the three design schemes. Note that both the sequential design technique and the single-mode concurrent design

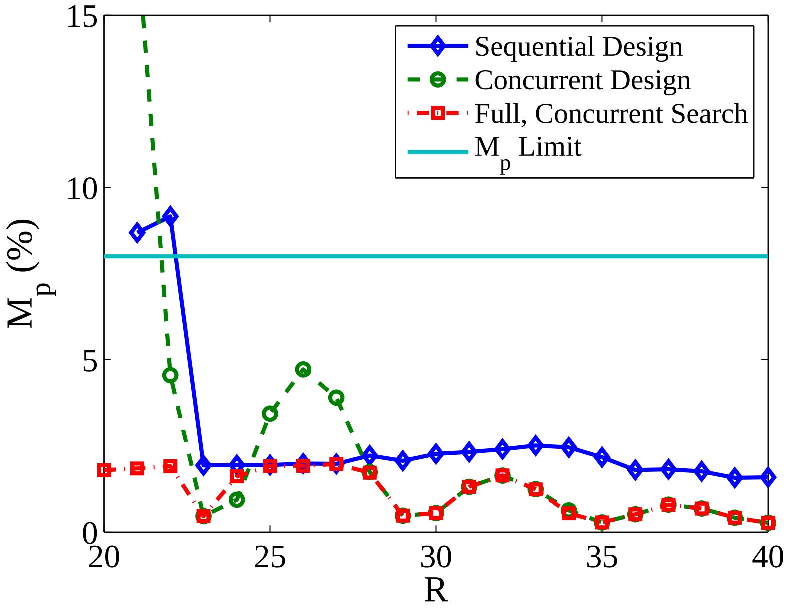


Figure 4.13: Percent Overshoot Comparison.

technique violate their overshoot constraint when R is low. This suggests that the error is mainly due to the single-mode assumption, and only partially due to inherent differences in the sequential and concurrent design strategies.

Figure 4.14 compares the disturbance rejection capabilities of each of the design strategies discussed here. Note that both concurrent design schemes are able to reject disturbances within the specified time. Only the sequential design technique occasionally fails to meet this specification.

Figure 4.15 shows that each of the design schemes is capable of maintaining its maximum actuator constraint. However, since speed was prioritized here, it was expected that each controller would have a maximum actuator effort close (if not equal to) the limit. One reason this does not occur is that the maximum actuator effort is a complicated function of the proportional gain, the DC gain term which amplifies the reference command (this DC gain is a function of K_p and approaches unity as $K_p \to \infty$), and the input shaper's impulse amplitudes. Another reason is that the concurrently designed controllers were limited to a maximum K_p of 600. If $K_p = 600$, then the DC gain is close to unity. And, if the input shaper's first

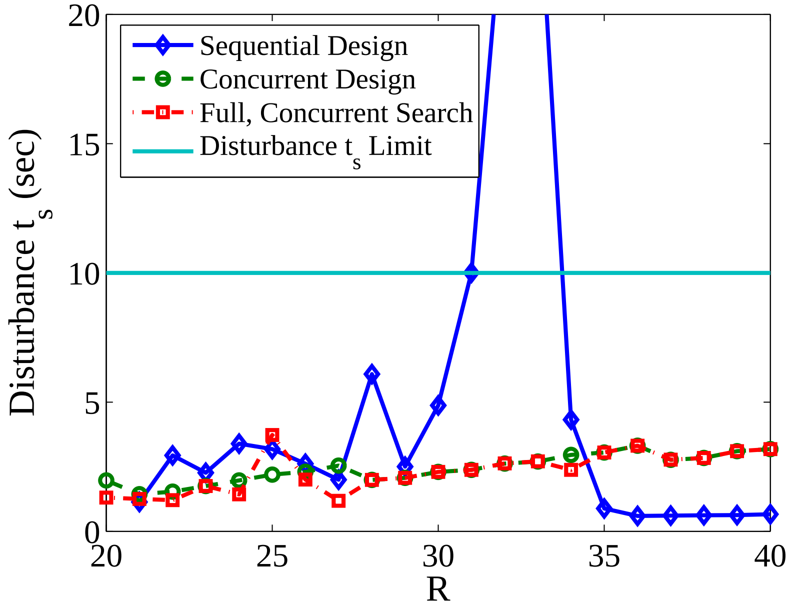


Figure 4.14: Disturbance Rejection Comparison.

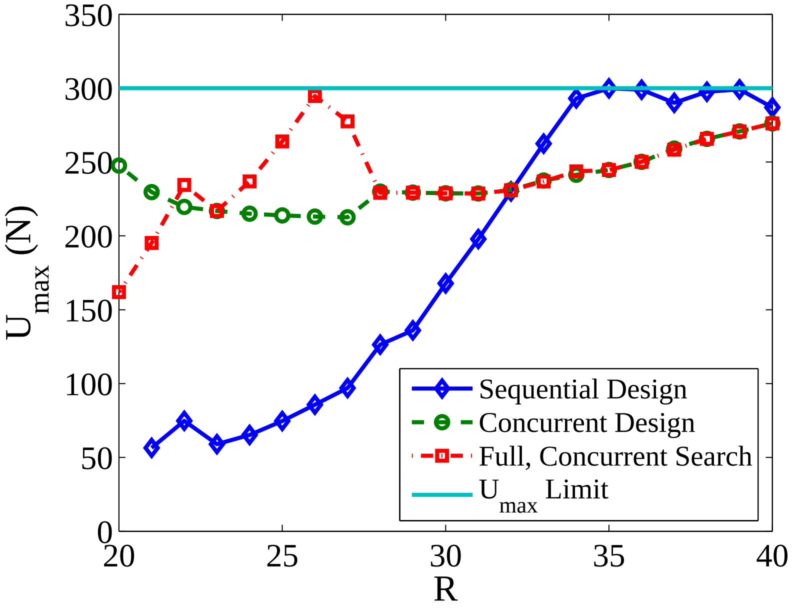


Figure 4.15: Maximum Actuator Effort Comparison.

impulse is less than 0.5, then the actuator effort at time $t = 0$ (this is often when U_{max} occurs) could be below 300. Remember that with ZVD and EI shapers, the first impulse amplitude is usually less than 0.5. The main reason that the sequential design technique seldom utilizes its maximum actuator effort is that this design technique naturally chooses smaller K_p values. This can be seen in Figure 4.16.

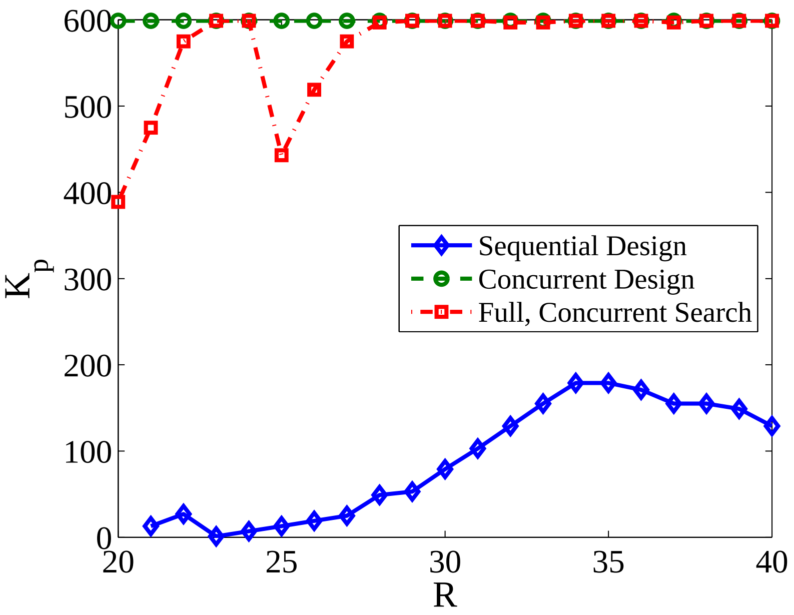


Figure 4.16: Kp Solution Comparison.

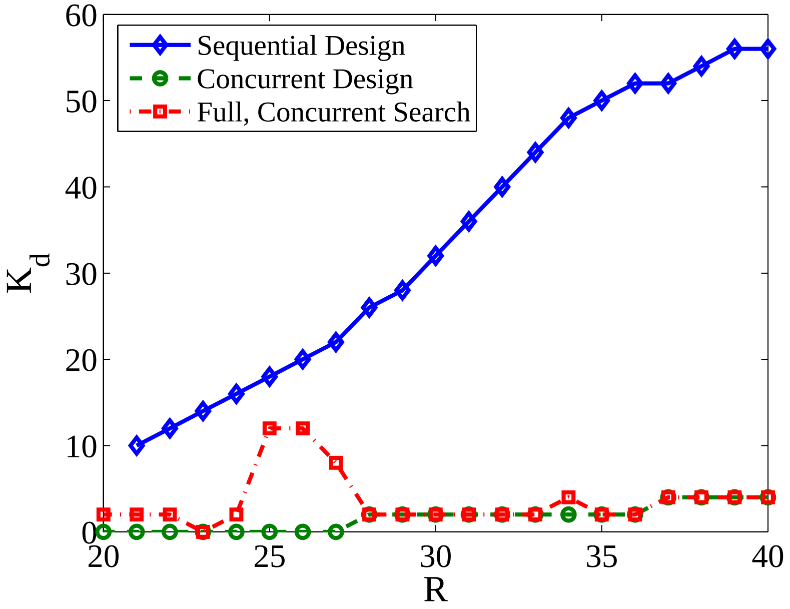


Figure 4.17: Kd Solution Comparison.

Figures 4.16 and 4.17 compare the proportional (K_p) and derivative (K_d) gains found by each of the design schemes. Notice that (as expected) both concurrent design techniques chose higher K_p and lower K_d gains than the sequential design technique. This is exactly how a concurrent design strategy works. Because input shaping is

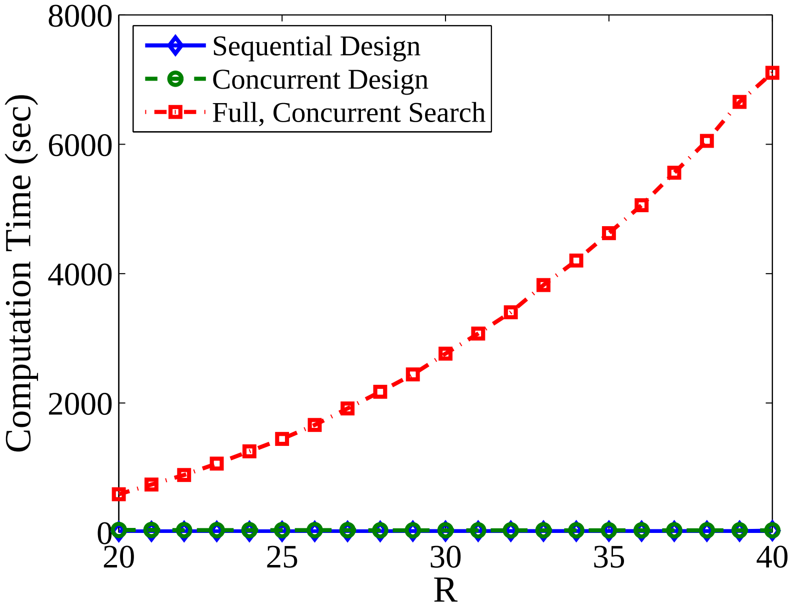


Figure 4.18: Computation Time Comparison.

able to eliminate a large amount of vibration, feedback controllers can create more oscillatory systems. This allows for faster settling times than are generally achievable solely by tuning the feedback controller (a sequential design technique).

Finally, Figure 4.18 shows the calculation times for each of the three design strategies. It is clear from this figure that choosing the full-search concurrent design technique requires a significant computation penalty. At its worst, the full-search concurrent design routine requires 229 times as long to execute as the single-mode concurrent design routine. At its best, the full-search concurrent design routine is only 16.5 times as long. Figure 4.19 shows a zoomed in view of Figure 4.18. This figure shows that even the single-mode concurrent design routine takes longer to execute than the sequential design routine. This difference is due to fact that the single-mode concurrent design routine has a larger solution space to search through than the sequential design routine has. The search space is increased for this routine because the use of input shaping allows for higher K_p values than are possible with a sequential design technique. However, both routines consistently execute fast, with the single-mode concurrent design routine requiring $2 - 3.5$ times as long to execute.

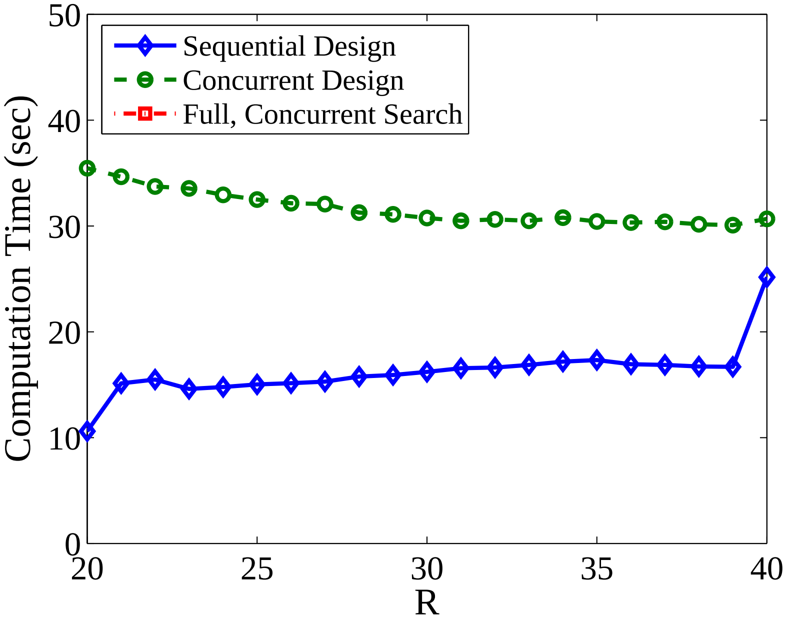


Figure 4.19: Computation Time Comparison - Zoomed in View.

4.3.1 Discussion of Single-Mode Approximation

This section illustrated that a single-mode approximation for a multi-mode system is often valid. This ability to approximate a complicated system as a simple one allows the use of quick and thorough concurrent design routines that will still be able to produce good controllers. In these studies, when the second mode was 12 times larger than the dominant mode ($R = 24$), the single-mode concurrent design routine yielded well-behaved controllers that outperformed a sequential design technique. Furthermore, when the second mode was greater than 15 times larger than the dominant mode $R = 30$, the single-mode concurrent design technique yielded excellent results that were equal to those of the full-search concurrent design routine. And, as shown in Figure 4.18, these high R values are when the computation time difference between concurrent design routines is at its highest. While this section only studied one, particular plant, this thesis will advocate the usage of a single-mode concurrent design routine when all higher modes are more than 15 times greater than the dominant mode.

It is important to note that the plant studied in this section was a particular, fourth-order system with one, left-half plane zero. The effects of additional oscillatory modes, first-order dynamics, and non-minimum phase plants were not addressed here. Each of these effects could change the frequency at which additional modes become non-negligible. Also, this study only showed the effect of the second mode becoming significant. The effect of the numerator becoming non-negligible was not addressed here. However, the study of these effects would have results similar to the study presented in this dissertation. When the extra dynamics are negligible, approximating the closed-loop system as a second-order oscillatory will allow for fast and reliable concurrent design procedures. However, as these extra dynamics become non-negligible, the single-mode approximation begins to break down. At this point, a more complicated concurrent design procedure will be necessary.

This section also illustrated a scenario where a multi-mode system could not appropriately be approximated as a single-mode system (when R was low). Unfortunately, many multi-mode systems fall into this category. This fact necessitates the advancement of concurrently designing OLIS and PID control to address plant types with multiple, non-negligible modes and numerator dynamics. The following chapter will address this issue, along with adding integral action to the feedback controller.

CHAPTER V

CONCURRENT DESIGN OF OLIS AND PID FEEDBACK CONTROL FOR GENERAL SYSTEMS

The three previous chapters have focussed on plants that are reasonably approximated as second-order systems with negligible numerator dynamics. For many real-world systems, this approximation works well. The experimental results on the portable bridge crane verify the usefulness of this simple modeling and control approach. However, there are also many scenarios where plants cannot be reasonably modeled as simple, second-order systems. For these applications, a new concurrent design methodology is needed. The block diagram studied in this chapter is shown in Figure 5.1. Here, the reference signal R is filtered by an outside-the-loop input shaper before entering a PID feedback controller (where the derivative action is implemented via velocity feedback).

There is currently no universally accepted design strategy for tuning PID controllers when the plant is more complicated than a simple, second-order system. This is because no current strategy is both feasible to implement and guaranteed to find

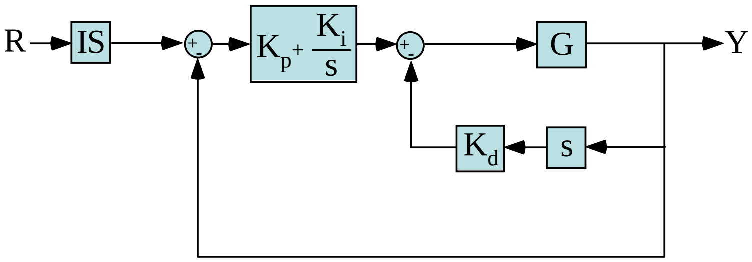


Figure 5.1: OLIS and PID Feedback Controller.

the optimal solution. Therefore, many PID controller design techniques exist, each with their own advantages and disadvantages.

This thesis will advance the state-of-the-art in concurrent design of outside-the-loop input shaping and PID feedback control in two ways. First, standard rules of thumb for designing PID controllers will be reformulated to work well with input shaping. Control engineers often use imprecise rules of thumb to design controllers. While certainly not guaranteeing an optimal solution, these rules of thumb produce good results with a minimal requirement in human and/or computer effort. This thesis will improve these rules by altering them to incorporate the effects and advantages of input shaping.

As a counterpoint to imprecise rules of thumb, controls engineers must sometimes use complicated and time consuming design techniques (often nonlinear, numerical optimizations) in order to push the performance limits and obtain near-optimal solutions. The second part of this chapter will develop a unique optimization technique for concurrently designing OLIS and PID control. This optimization technique will split a typically nonlinear minimization into a linear minimization combined with a simplified nonlinear minimization. Compared to a well-known nonlinear optimization package, this new technique yields superior and more-consistent results.

5.1 Tuning Rules for Combined OLIS/PID Feedback Control Schemes

5.1.1 Ziegler-Nichols Tuning Rules

The Ziegler-Nichols (Z-N) tuning rules are a well-known and accepted procedure by which to design PID feedback controllers [14, 55]. However, they were established before the development of input shaping. Therefore, they only address the design of the feedback system (they only choose the P, I and D gains). By adjusting these rules to incorporate the effect and design of outside-the-loop input shapers, superior control systems can be created. This will first be demonstrated by comparing the results of a Z-N based design process to those of a design process based upon the newly proposed OLIS/PID tuning rules. Note that the difference between these two design processes (Z-N and OLIS/PID tuning rules) is parallel to the difference between the sequential and concurrent design of OLIS and PID controllers as discussed in the previous three chapters. The Z-N tuning rules first choose the PID gains without regard to any input shaping effects. Only then is an input shaper added to further cancel any remaining, oscillatory dynamics. The OLIS/PID tuning rules proposed here specifically choose the PID gains based upon the knowledge that an input shaper will be added to the control system for the purpose of vibration reduction.

Two textbook Z-N tuning rules examples will be reviewed here to demonstrate the superiority of the newly proposed OLIS/PID tuning rules. In both of these examples, the textbooks follow the prescribed Z-N tuning rules to initially develop a PID controller. However, the overshoot resulting from the accepted tuning rules proved too high. So, undetailed "tweaking" was performed by the authors (Ogata and Franklin) to adjust the gains so that acceptable overshoot was obtained. The OLIS/PID tuning rules developed here will intentionally create a closed-loop system with more vibration than is desired in the step response. The resulting system will have acceptable reference responses due to the use of the input shaper. And, because the closed-loop system is left more oscillatory than when using the Z-N tuning rules, the overall control system developed by the new OLIS/PID tuning rules will be faster than control systems developed by strictly following the Z-N design process.

5.1.1.1 Ziegler-Nichols Design Process: Example 1

The first example was taken from Ogata's textbook, "System Dynamics" [55]. The plant and PID controller are:

Plant:

$$G = \frac{1}{s(s+1)(s+5)} \tag{5.1}$$

PID form:

$$C = Kp\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{5.2}$$

The open-loop step response can be seen in Figure 5.2.

Z-N Design - Step 1: Find K_p that results in marginal stability and period of resulting oscillations (P_{cr}). These values are given as $K_{p,cr} = 30$ and $P_{cr} = 2.81$. According to Z-N tuning rules, the desired PID gains are $K_p = 18$, $T_i = 1.405$ and $T_d = 0.35124$. This control scheme results in the step response shown in Figure 5.3. Because the desired overshoot limit was 25%, adjustments had to be made.

Z-N Design - Step 2: Fine tune PID gains to achieve desired overshoot. The new gains were given as $K_p = 18$, $T_i = 3.077$ and $T_d = 0.7692$. Note that the Ogata textbook does not describe the fine-tuning rules used to adjust the gains. The resulting step response is shown Figure 5.4. The overshoot is now at an acceptable level.

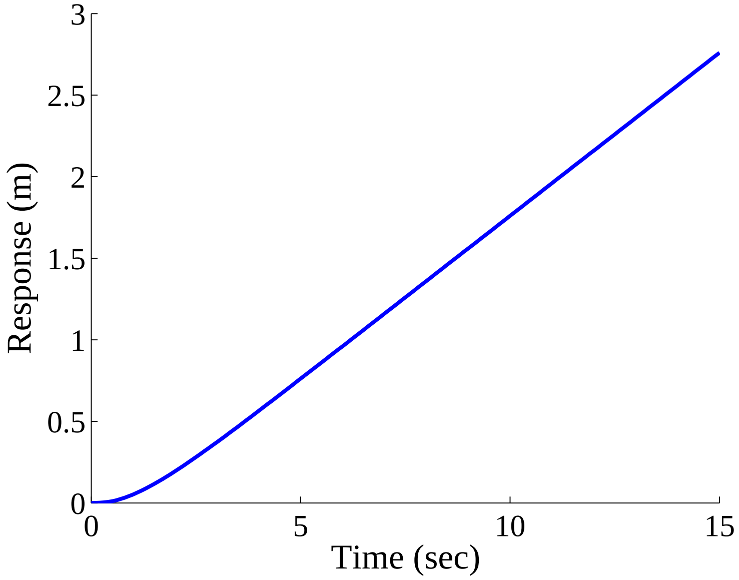


Figure 5.2: Z-N Example #1: Open-Loop Step Response of G.

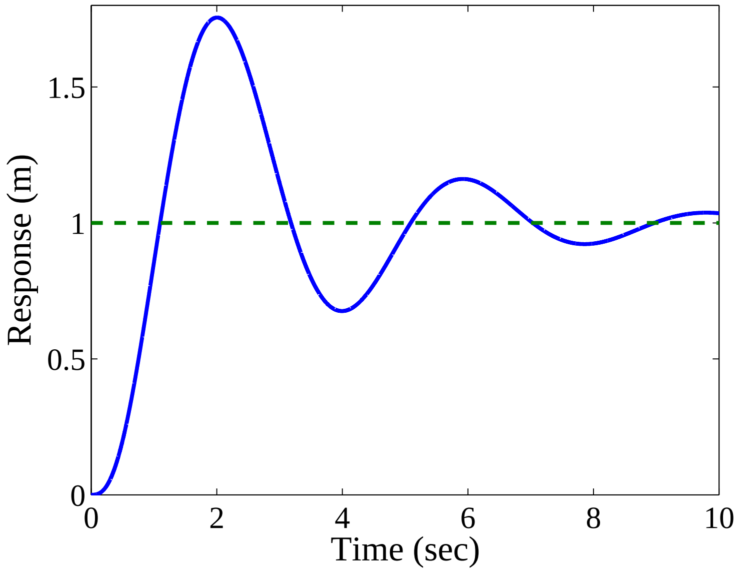


Figure 5.3: Z-N Example #1: Step 1 Result.

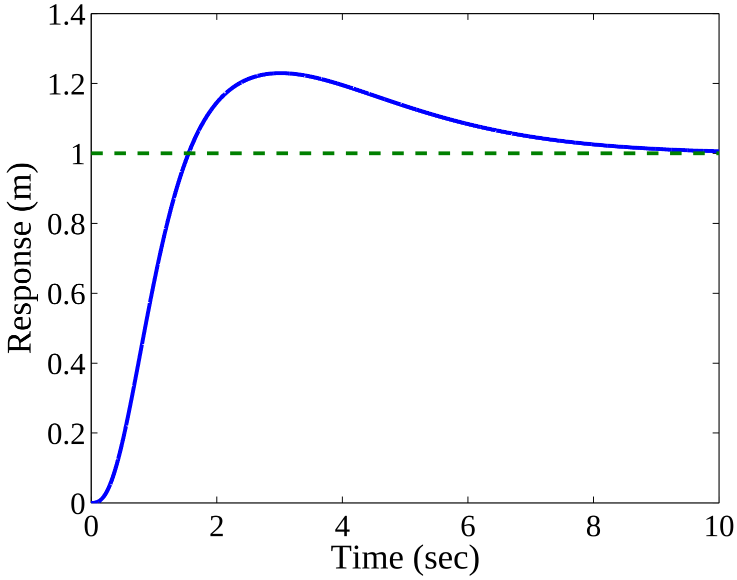


Figure 5.4: Z-N Example #1: Step 2 Result.

Z-N Design - Step 3: Addition of outside-the-loop input shaping. Because the Z-N method with auxiliary tweaking by Ogata produced such high damping ($\zeta = 0.67$), input shaping is of little help in terms of further reducing the overshoot (the vibration resulting from such highly damped dynamics is very small). However, the appropriate ZV shaper is $A_1 = 0.95$, $A_2 = 0.05$ and $t_2 = 1.21$. The resulting OLIS/PID step

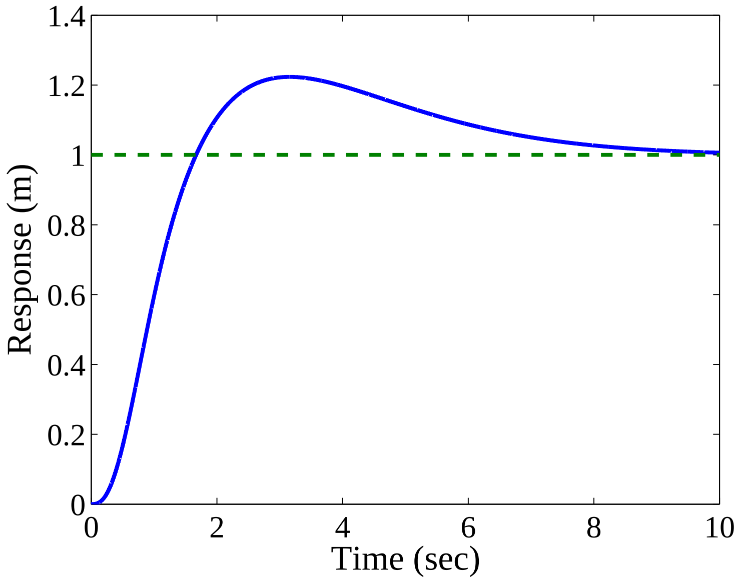


Figure 5.5: Z-N Example #1: Step 3 Result.

response is shown in Figure 5.5. Note that the overshoot is only slightly improved over Step 2. The overshoot is mainly a result of the closed-loop system's numerator dynamics.

5.1.1.2 Ziegler-Nichols Design Process: Example 2

The second example comes from Franklin, Powell and Emami-Naeini's book "Feedback Control of Dynamic Systems" and utilizes Ziegler-Nichols first design method [14]. The plant is intended to model a heat exchanger, but the actual plant parameters are not well defined by the book. From the basic structure and plots shown, G was estimated to be:

Plant:

$$G = \frac{e^{-10s}}{70s + 1} \tag{5.3}$$

PI form: This example only uses PI control.

$$C = K_p \left(1 + \frac{1}{T_i s}\right) \tag{5.4}$$

The open-loop step response can be seen in Figure 5.6.

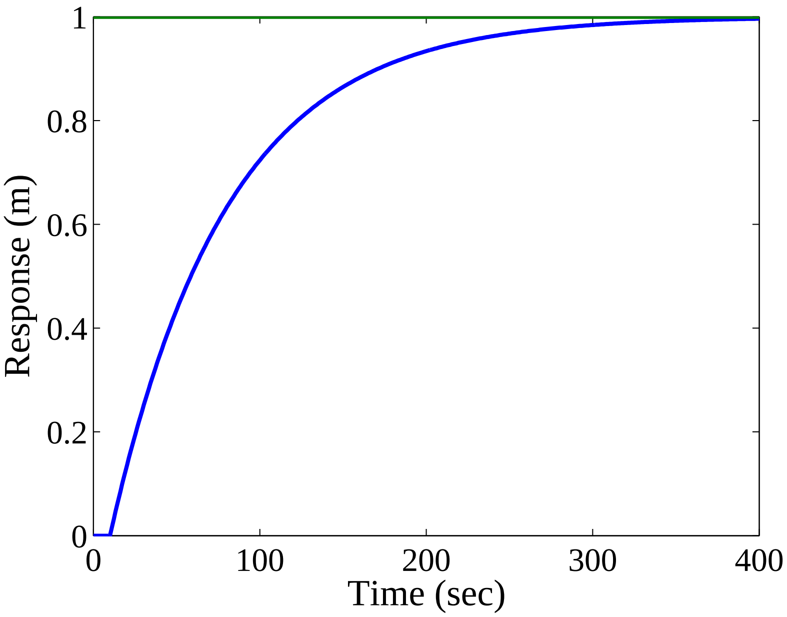


Figure 5.6: Z-N Example #2: Open-Loop Step Response of G.

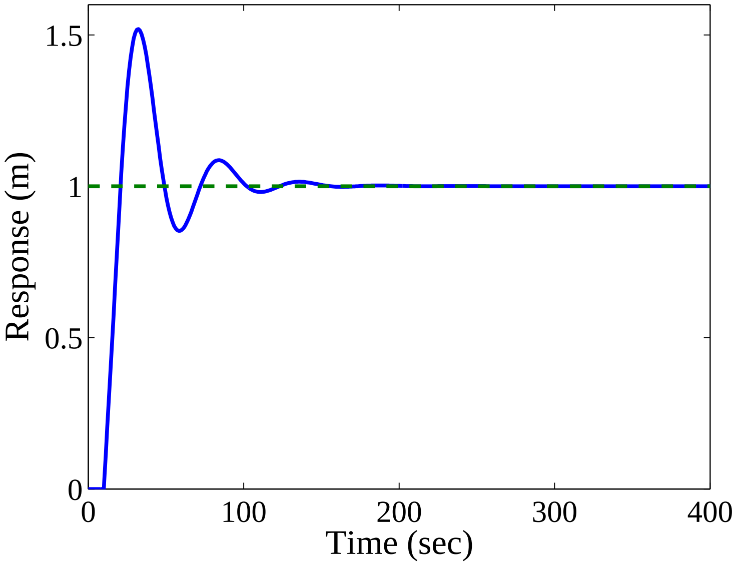


Figure 5.7: Z-N Example #2: Step 1 Result.

Z-N Design - Step 1: Find maximum slope (R) of open-loop response ($R = \frac{1}{90}$) and the system lag, L ($L = 13sec$). According to Z-N tuning rules, $K_p = 6.22$ and $T_i = 43.3$. This control scheme results in the step response shown in Figure 5.7. Because the overshoot is too high (the desired overshoot was less than 25%), adjustments were made.

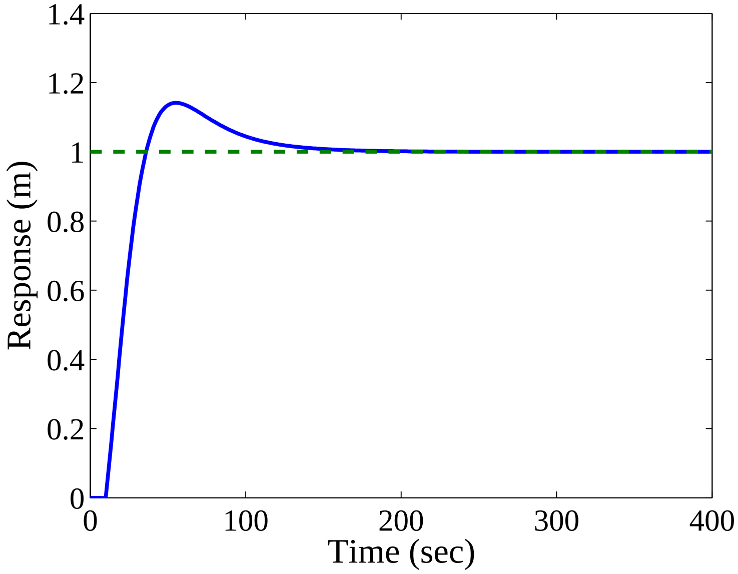


Figure 5.8: Z-N Example #2: Step 2 Result.

Z-N Design - Step 2: Fine tune PID gains to achieve better overshoot. As in the first Z-N example, the "tweaking" process was not well defined by Franklin, Powell and Emami-Naeini. The new gains were chosen to be $K_p = 3.11$ and $T_i = 43.3$. The resulting step response is shown Figure 5.8. The overshoot is now at an acceptable level.

Z-N Design - Step 3: Addition of input shaping. Because the plant is not a rational transfer function, a numerical root-locus technique was employed to find the dominant poles. Again, the Z-N method with additional "tweaking" resulted in high damping ($\zeta = 0.76$), and input shaping is of little help. However, the appropriate ZV shaper is $A_1 = 0.98$, $A_2 = 0.02$ and $t_2 = 52.4$. The resulting step response is shown in Figure 5.9. Note that the overshoot is only slightly improved over Step 2. The overshoot is again primarily a result of the closed-loop system's numerator dynamics.

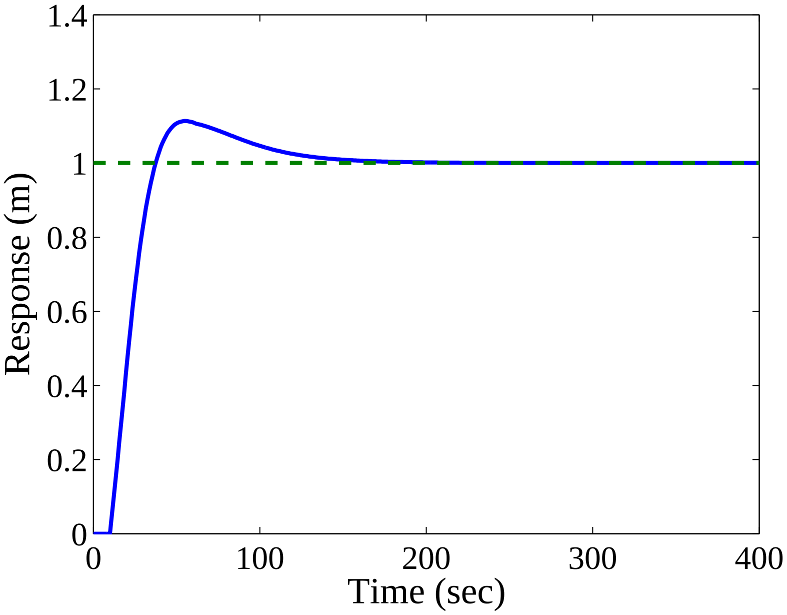


Figure 5.9: Z-N Example #2: Step 3 Result.

5.1.2 New Tuning Rules for OLIS/PID Feedback Controller Combinations

Based on the known effects of input shaping and feedback control, a new set of tuning rules is developed for use on controllers that combine outside-the-loop input shaping and PID feedback control. Similar to the Z-N tuning rules, the new OLIS/PID tuning rules seek to achieve an overall controller that yields less than 25% overshoot. However, the new approach has one major difference. The OLIS/PID tuning rules create a closed-loop system with more than 25% overshoot in response to a step input. However, because the OLIS/PID tuning rules consider the addition of an input shaper, a more oscillatory system can confidently be designed. The input shaper will cancel most of the natural oscillations, resulting in an overall controller that yields approximately 25% overshoot (or less) and little residual oscillation.

The new tuning rules are similar to a concurrent design of OLIS and PID feedback controllers because the choice of PID gains is dependant upon the known effects of input shaping. The new OLIS/PID tuning rules will generally produce controllers

Table 5.1: New OLIS/PID Tuning Rules - First Method

	K_p	$\frac{K_p}{K_i}$	$\frac{K_d}{K_p}$
P	$\frac{1.1}{RL}$	∞	0
PI	$\frac{1}{RL}$	$\frac{L}{0.4}$	0
PID	$\frac{1.3}{RL}$	$1.9L$	$0.4L$

Table 5.2: New OLIS/PID Tuning Rules - Second Method

	K_p	$\frac{K_p}{K_i}$	$\frac{K_d}{K_p}$
P	$0.6K_{cr}$	∞	0
PI	$0.5K_{cr}$	$\frac{P_{cr}}{1.5}$	0
PID	$0.7K_{cr}$	$0.4P_{cr}$	$0.1P_{cr}$

superior to those that rely on the Z-N tuning rules (which choose the PID gains without regard to the effects of input shaping) followed by an input shaper.

Similar to the Z-N tuning rules, the newly proposed tuning rules are split into two "methods" depending upon the form of the plant. "Method #1" is for plants without integrators or dominant complex-conjugate poles. "Method #2" is for plants that will exhibit sustained oscillations under proportional control for some proportional gain (K_p). The tuning rules for "Method #1" are given in Table 5.1. Here, R is the slope of the plant's step response and L is the lag inherent to the step response of the plant. The tuning rules for "Method #2" are given in Table 5.2. Here, K_{cr} is the proportional gain that would cause instability if the plant was controlled solely with proportional control. P_{cr} is the period of that marginally stable system. Note that for both methods, the newly proposed gains are more aggressive versions of those proposed by Ziegler and Nichols. The K_p and K_i gains are slightly larger and the K_d gains are slightly smaller.

The newly proposed OLIS/PID tuning rules share an important advantage with

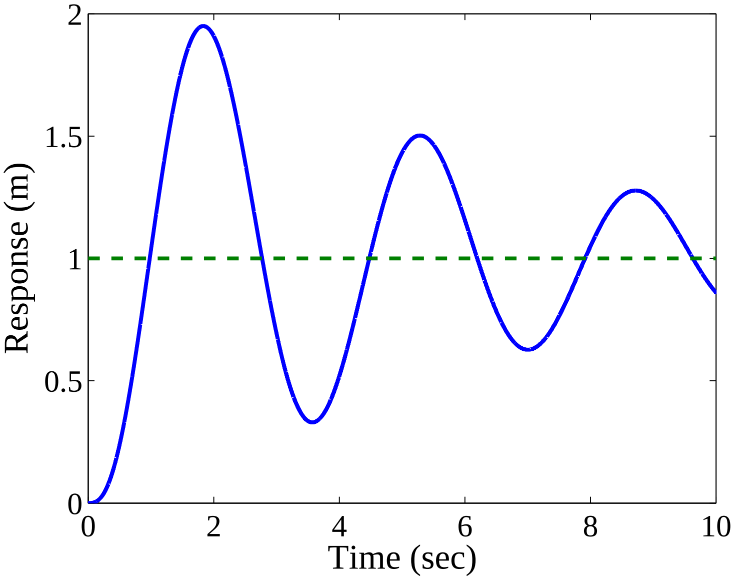


Figure 5.10: Example #1 - OLIS/PID Tuning Rules - Unshaped Step Response.

the Ziegler-Nichols tuning rules. Because they are based upon the system response, the design process can be carried out experimentally. That is, no specific plant model is required.

The following two sections will revisit the controller design processes detailed in Sections 5.1.1.1 and 5.1.1.2. However, the new OLIS/PID tuning rules will be used, as opposed to the Z-N tuning rules.

5.1.2.1 OLIS/PID Design Process: Example 1

OLIS/PID Design - Step 1: This first step after identifying the plant parameters is to use the newly-proposed OLIS/PID tuning rules. Because this plant requires the use of “Method #2”, the OLIS/PID tuning rules are given in Table 5.2. Remember from Section 5.1.1.1 that $K_{p,cr} = 30$ and $P_{cr} = 2.81$. According to the newly proposed OLIS/PID tuning rules, $K_p = 21$, $K_d = 5.9$, and $K_i = 18.7$. These PID gains translate into $K_p = 21$, $T_d = 0.28$, and $T_i = 1.12$. The unshaped step response of the closed-loop system under these gains is shown in Figure 5.10. Note that this a more oscillatory closed-loop system than was designed by the Z-N tuning rules (Figure 5.3).

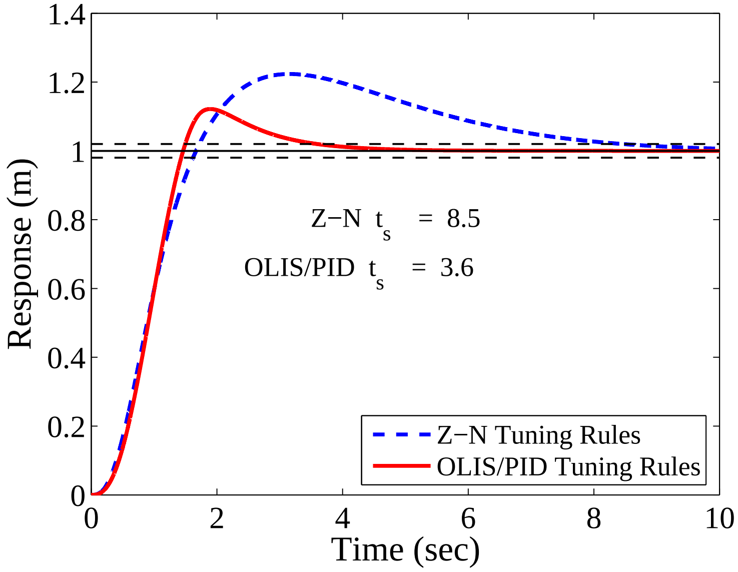


Figure 5.11: Example #1 Step Response Comparison.

OLIS/PID Design - Step 2: The next step is to apply the input shaper. As in each of the above examples, a ZV shaper was chosen. When the input shaper is added to the PID feedback controller, the overall system has the step response that is shown in Figure 5.11. This figure shows that the system designed via the OLIS/PID tuning rules has less overshoot and is twice as fast (in terms of settling time) as the system design via the Z-N tuning rules.

5.1.2.2 OLIS/PID Design Process: Example 2

OLIS/PID Design - Step 1: The plant used in Example #2 requires the use of “Method #1”. The OLIS/PID tuning rules for “Method #1” are given in Table 5.1. Remember from Section 5.1.1.2 that $R = \frac{1}{90}$ and $L = 13sec$. Therefore, the PI gains used in the feedback controller were $K_p = 6.92$ and $K_i = 0.213$ (or $T_i = 32.5$). The unshaped step response of the closed-loop system under these gains is shown in Figure 5.12. Again, this a more oscillatory closed-loop system than was designed by the Z-N tuning rules (Figure 5.7).

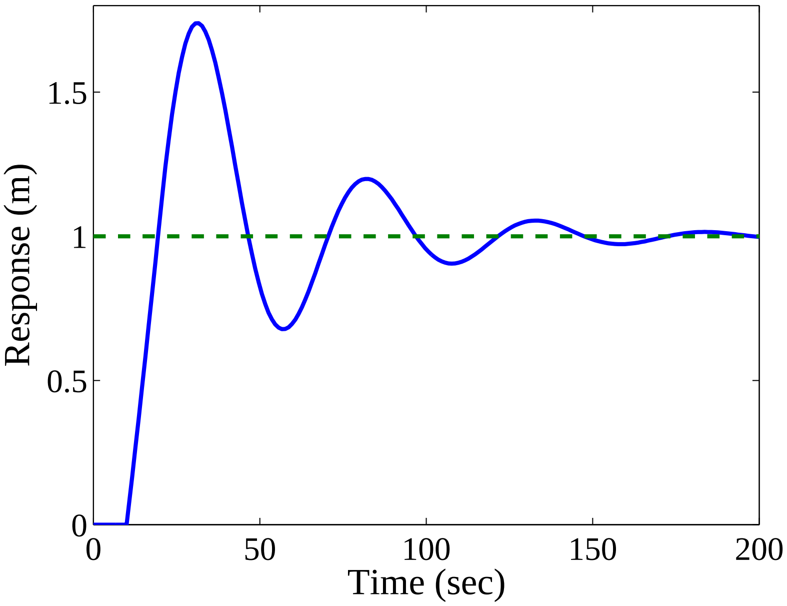


Figure 5.12: Example #2 - OLIS/PID Tuning Rules - Unshaped Step Response.

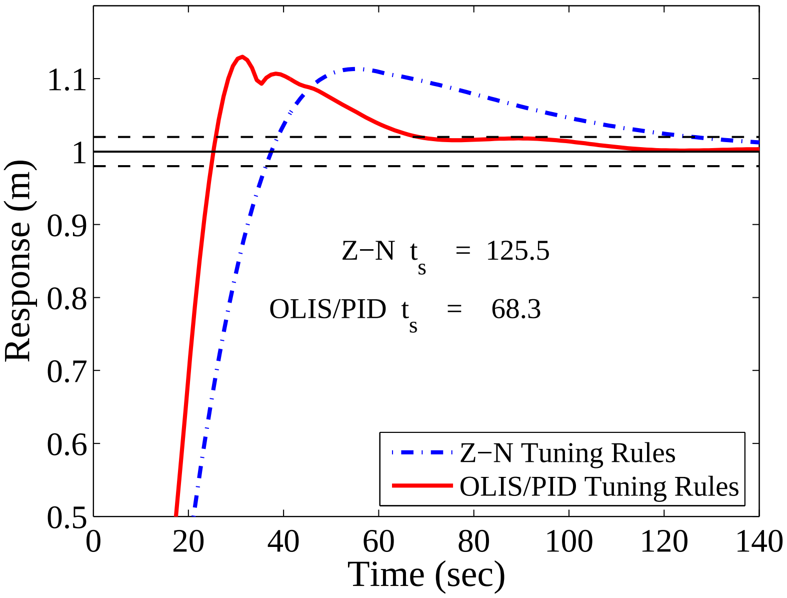


Figure 5.13: Example #2 Step Response Comparison.

OLIS/PID Design - Step 2: The next step is to apply the input shaper. When the input shaper is added to the PID feedback controller, the overall system has the step response that is shown in Figure 5.13. This figure shows that the system designed via the OLIS/PID tuning rules is, again, approximately twice as fast as the system design via the Z-N tuning rules. However, in this case, the system designed via the OLIS/PID tuning rules has a slightly higher overshoot (even though both controllers are well under the 25% overshoot goal).

5.1.2.3 Fine Tuning Process

As with any set of tuning rules that are used on a wide variety of plants, the exact performance characteristics (overshoot, damping, speed, etc.) cannot be specified or known during the design process. The tuning rules are used and accepted because they give reasonably good performance for a large number of plants. However, there will always be scenarios when the tuning rules fail to yield an acceptable control system. This was seen in Step #1 of both Ziegler-Nichols design examples, where the overshoot resulting from using the Z-N tuning rules was above the desired 25%. Therefore, there will always need to be a procedure for adjusting the controller gains until an acceptable response is obtained. Unfortunately, the textbooks used to highlight the Z-N design process used undocumented "tweaking" to adjust the PID gains and achieve less than 25% overshoot. This dissertation will present a precise method for adjusting the PID gains in the case that the OLIS/PID design procedure does not yield an acceptable step response.

Because the OLIS/PID tuning rules produce more aggressive systems, the most likely problem will be that the overall controller has too much overshoot, or that the closed-loop system has too little damping (important for disturbance rejection - which is not aided by the outside-the-loop input shaper), or that the controller has unacceptably small stability margins. Fortunately, each of these problems can usually be solved in the same manner: make the closed-loop system less aggressive.

The fine tuning process for the design procedure based upon the newly proposed OLIS/PID tuning rules is as follows:

1. *P Controller:* Reduce K_p by 10%.

2. *PI Controller:* Reduce K_p and K_i by 5%.

3. *PID Controller:* Reduce K_p and K_i by 3% and Increase K_d by 3%.

These steps can be performed multiple times, until an acceptable OLIS/PID response is obtained. Note, however, that adjusting the PID gains changes the closed-loop poles. Therefore, each time the PID gains are adjusted, the input shaper will need to be redesigned.

5.1.3 Frequency Domain Design - Bode Diagram

Another common feedback controller design method analyzes the system via the Bode diagram. This section will give an example of such a design, again comparing sequential and concurrent design of OLIS and feedback control. Here, the bandwidth (ω_b) will be used as an approximate measure of the closed-loop system's speed (rise time or peak time). The open-loop system's phase margin (PM) will be used as an approximate measure of the closed-loop system's damping (ζ) and overshoot. These performance approximations are taken from [14, 55].

5.1.3.1 Example - PI control of a third-order plant

Plant:

$$G = \frac{s+1}{(s+3)(s^2+0.1s+2)} \tag{5.5}$$

PI form:

$$C = K_p + \frac{K_i}{s} \tag{5.6}$$

The goal of the design procedures described here will be to create the fastest possible step response while limiting overshoot to less than 50%. Remember that the sequential design procedure only uses the PID feedback controller to limit overshoot. The concurrent design procedure will choose the PID gains with the vibration reducing capabilities of input shaping in mind. Consequently, the concurrently designed

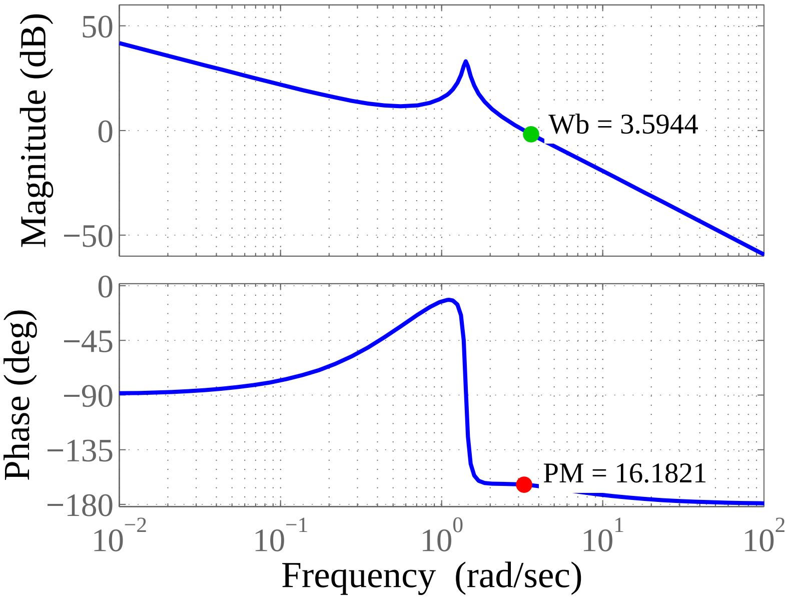


Figure 5.14: Sequential Design Bode Diagram.

PID feedback controller will have a smaller phase margin. Fortunately, this will allow for a larger bandwidth. The result will be on overall faster control system with acceptable vibration suppression.

Sequential Design: Using MATLAB's "sisotool", the gains were chosen to be $K_p = 10.95$ and $K_i = 7.3$. The resulting Bode diagram is shown in Figure 5.14. The step response of the resulting closed-loop system is shown in Figure 5.15. This figure shows that the overshoot constraint has been met by the PID feedback controller, while maintaining a fast rise time.

The resulting closed-loop system has a damping ratio of 0.13. Therefore, the complimentary ZV shaper is defined by $A_1 = 0.6$, $A_2 = 0.4$ and $t_2 = 0.97$. The resulting step response is shown in Figure 5.16.

Concurrent Design: Again using MATLAB's "sisotool", the gains were chosen to be $K_p = 55.8$ and $K_i = 37.2$. The resulting Bode diagram is shown in Figure 5.17. Note that this system has a higher bandwidth. The step response of the resulting closed-loop system is shown in Figure 5.18. Note that this response is faster, but has

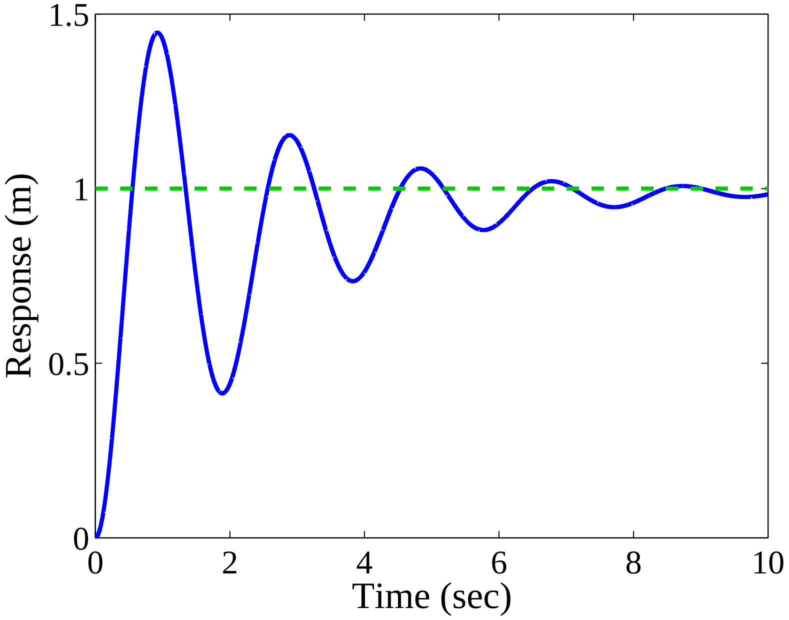


Figure 5.15: Sequential Design Step Response - Without Input Shaper.

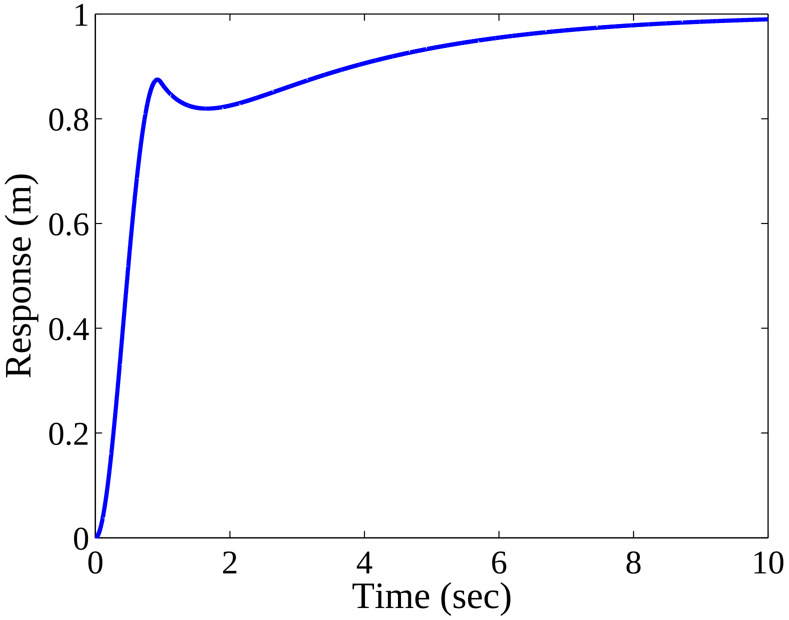


Figure 5.16: Sequential Design Step Response - With Input Shaper.

an unacceptable amount of overshoot. Fortunately, this overshoot problem will be negated by the input shaper.

The resulting closed-loop system had a damping ratio of 0.09. Therefore, the complimentary ZV shaper is defined by $A_1 = 0.57$, $A_2 = 0.43$ and $t_2 = 0.43$. The resulting step response is shown in Figure 5.19.

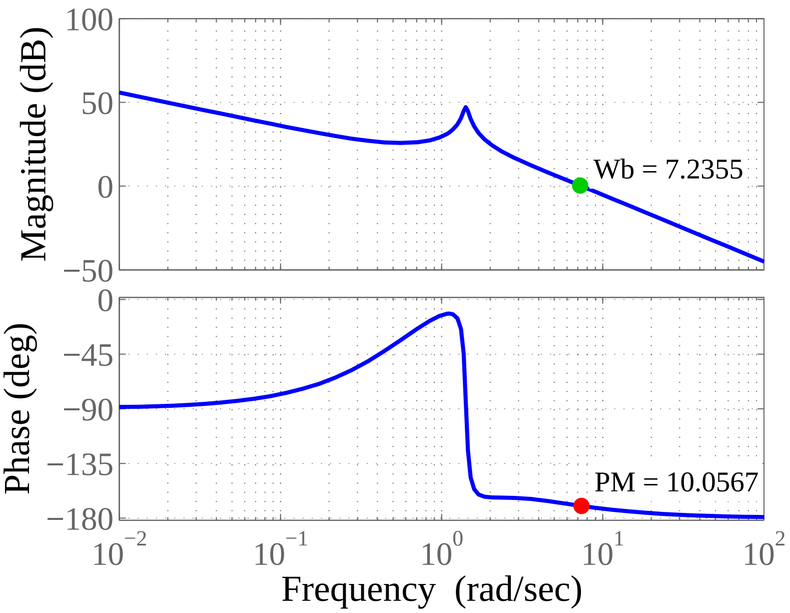


Figure 5.17: Concurrent Design Bode Diagram.

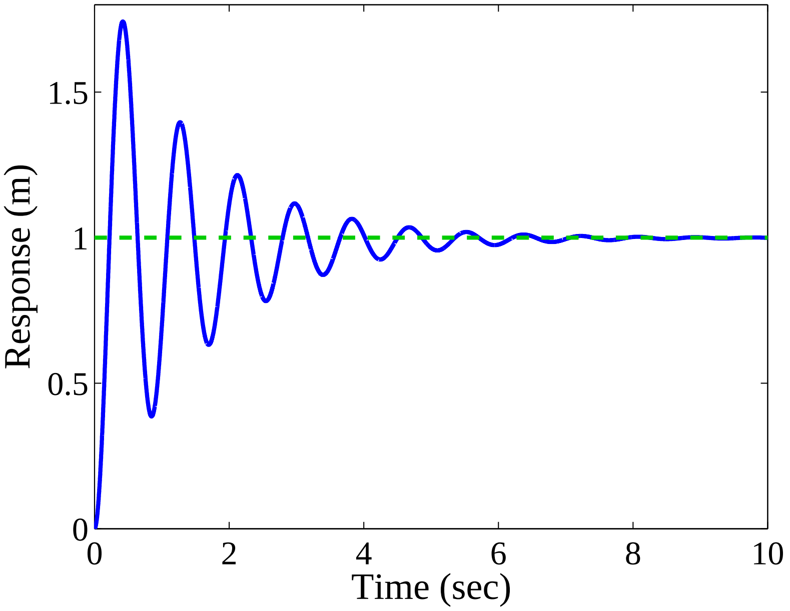


Figure 5.18: Concurrent Design Step Response - Without Input Shaper.

Figure 5.20 compares the step responses of the sequentially and concurrently designed control systems. Clearly, the concurrently designed control scheme has a much faster response in terms of both rise time and settling time. However, as indicated by Figure 5.21, the cost is a more than three-fold increase in the maximum actuator

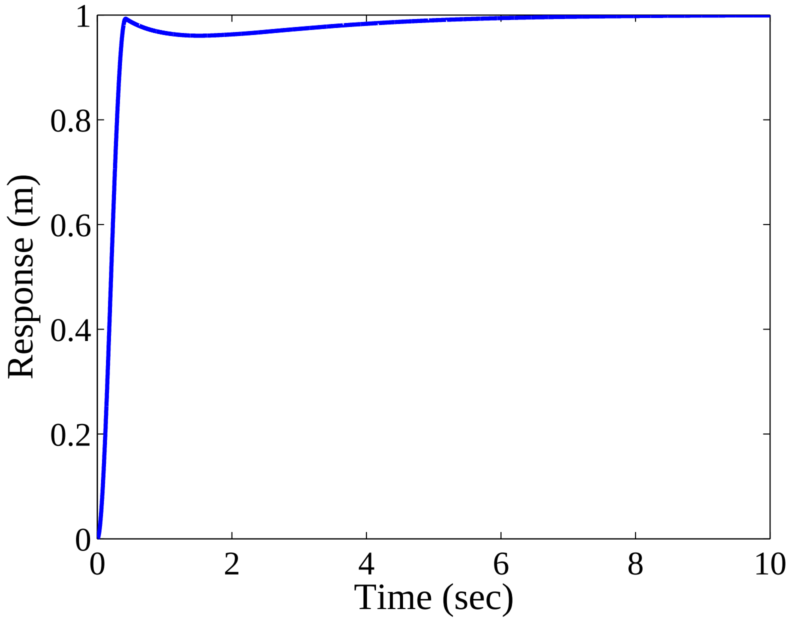


Figure 5.19: Concurrent Design Step Response - With Input Shaper.

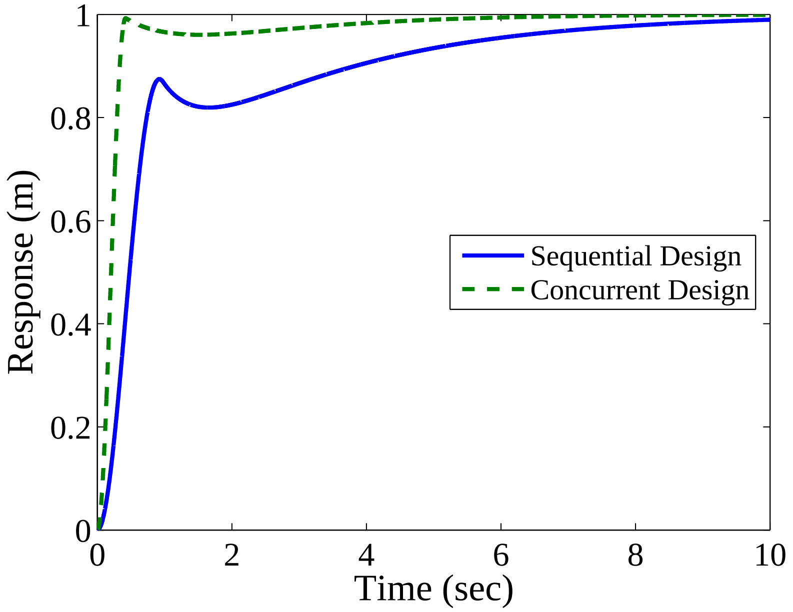


Figure 5.20: Step Response Comparison.

effort. This increase in maximum actuator effort could be too demanding for a practical control scheme. Therefore, actuator effort should be taken into account when designing these controllers. Because bandwidth is the measure of speed, it should also be an approximate measure of actuator effort - the faster a system moves, the more effort will be required to move it.

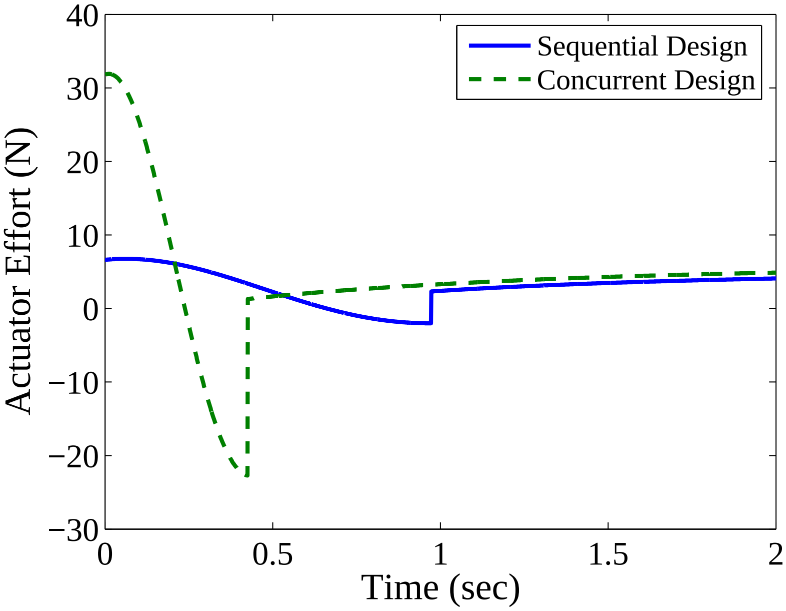


Figure 5.21: Actuator Effort Comparison.

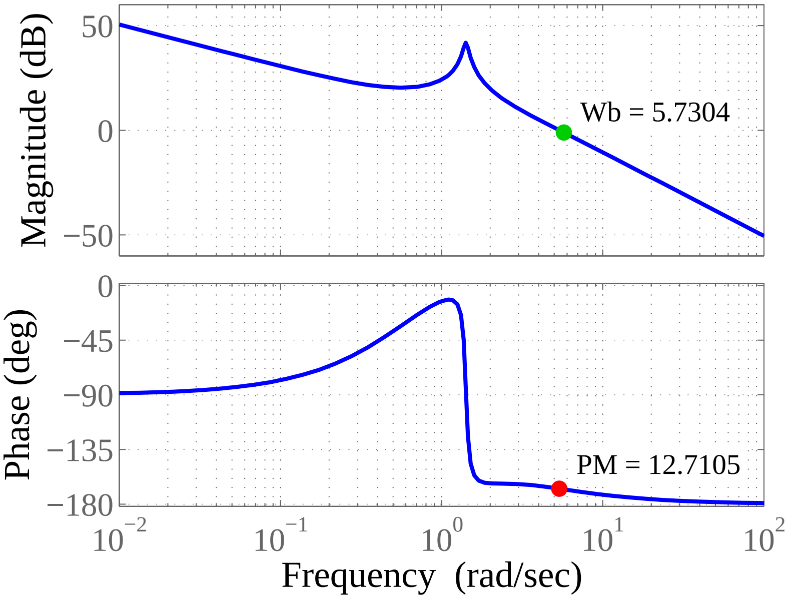


Figure 5.22: Bode Diagram of Modified Concurrent Design.

The concurrent design scheme can be redone with a compromise on bandwidth. Figure 5.22 shows a Bode diagram of a new open-loop system. Notice that it has a bandwidth higher than the sequentially designed control scheme and lower than the original, concurrently designed system. Likewise, the phase margin of the modified,

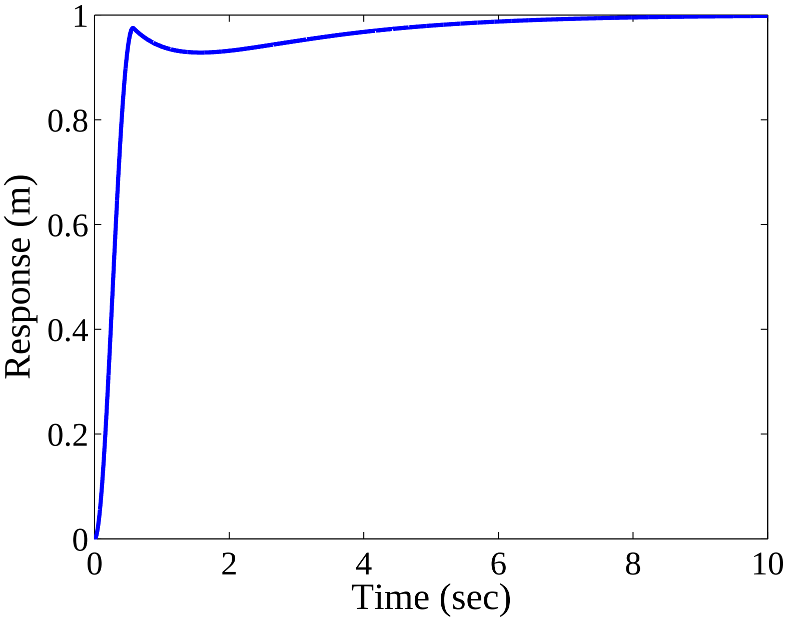


Figure 5.23: Step Response of Modified Concurrent Design.

concurrently designed controller is between the values of the sequentially designed controller and the original, concurrently designed controller. The step response of this modified, concurrently designed control scheme is shown in Figure 5.23 and is labeled "Concurrent Design (lower U_{max})".

Comparison of all three control schemes in Figure 5.24 shows that the concurrent design procedure can compromise between speed and actuator effort while still meeting constraints and outperforming a sequentially designed controller (in terms of speed). Figure 5.25 shows the actuator requirements of all three control schemes. Notice that the second concurrently designed controller requires just over one half the maximum actuator effort as the first concurrently designed controller. However, the second concurrently designed controller still has a significantly faster step response, as compared to the sequentially designed control scheme.

5.1.3.2 Loop Shaping

The more rigorous extension of the previously described frequency domain design procedure is loop shaping. As described in Chapter 1, loop shaping is a well-known

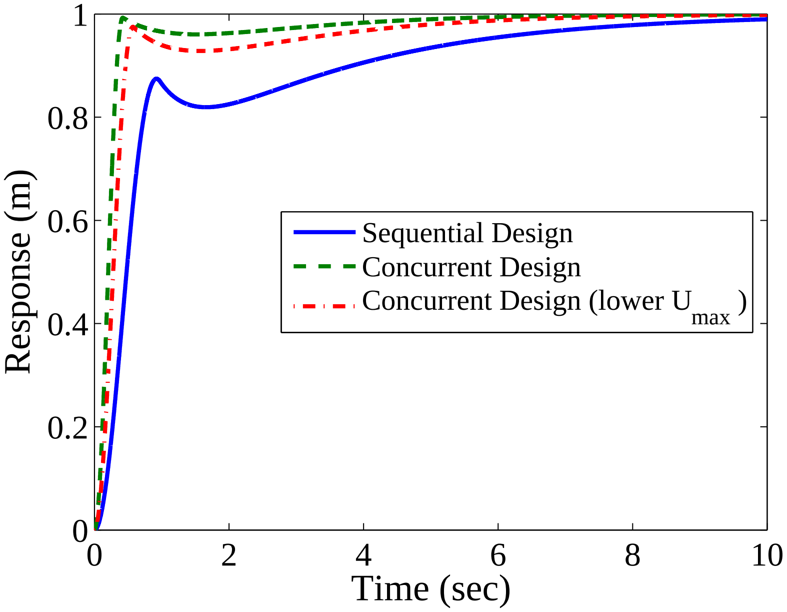


Figure 5.24: Step Response Comparison of all Three Designs.

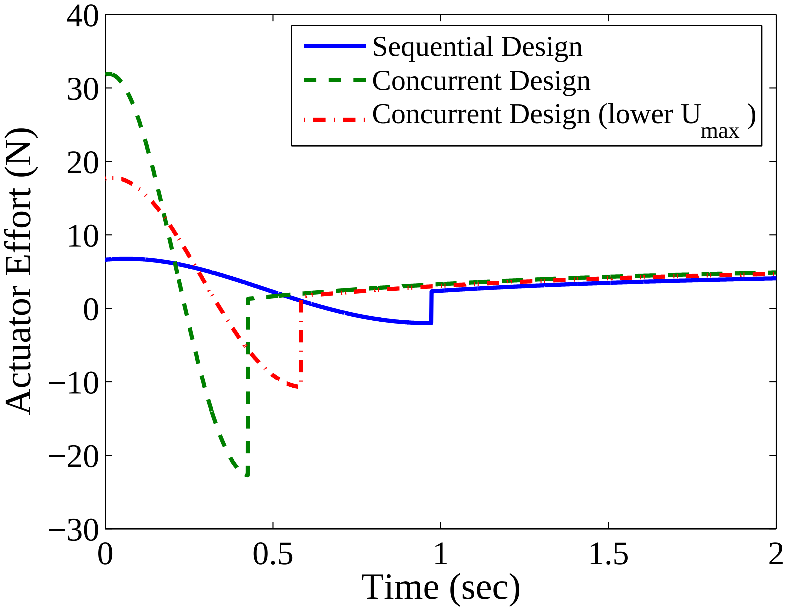


Figure 5.25: Actuator Effort Comparison of all Three Designs.

frequency domain design technique. Loop shaping begins by establishing a desired Bode plot (or at least a desired frequency response for some finite range of frequencies). Then, various optimization techniques can be used to create a controller that, combined with the plant, yields an open-loop transfer function whose actual Bode plot is close to the desired Bode plot. Of course, the desired Bode plot is chosen by the engineer for its ability to achieve certain performance criteria, while not violating various constraints (such as actuator effort limitations).

Any of the accepted, state-of-the-art loop shaping techniques can be modified for use within a concurrent OLIS/PID design scheme. Similar to the frequency domain example given above, the rules for what constitutes an "acceptable" Bode plot would change in a concurrent design scheme. The closed-loop system can have more overshoot (less phase margin) and higher actuator efforts.

5.2 Rigorous Optimization Technique for use on Generic Plants

The previous section described a concurrent design technique based on general guidelines and principles. However, there are many circumstances when more precise design techniques are required. While usually requiring more knowledge, effort and computation time, these rigorous design techniques will generally produce superior solutions. This section will present a numeric optimization technique for concurrently designing OLIS/PID controllers for use on any plant that can be stabilized via PID control. The optimization technique will minimize the system's settling time to a desired, step-reference command. Settling time will be minimized subject to the following constraints: the closed-loop system must be stable, the slowest time constant must be above a minimum value (for disturbance rejection), the system's overshoot must remain below a certain level, the maximum actuator effort must remain below a certain level.

5.2.1 Solution Routine Description

A typical nonlinear optimization routine for concurrently designing OLIS/PID controllers is depicted in Figure 5.26. This solution routine utilizes MATLAB's nonlinear optimization technique named "fmincon". This standard optimization package

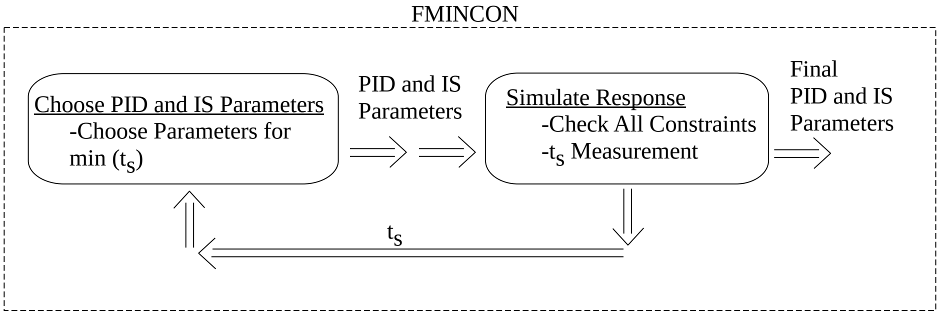


Figure 5.26: Fully-Nonlinear Optimization Routine.

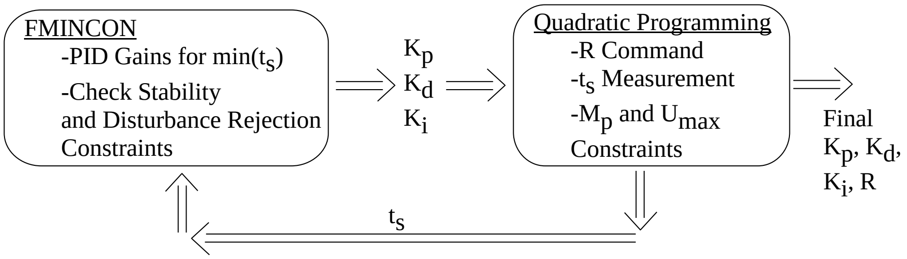

FMINCON
-PID Gains for min(t_s)
-Check Stability and Disturbance Rejection Constraints
K_p
K_d
K_i
Quadratic Programming
-R Command
-t_s Measurement
-M_p and U_{max} Constraints
Final K_p, K_d, K_i, R
t_s

Figure 5.27: Partially-Linear Optimization Routine.

chooses a set of input shaper and feedback parameters. Then, the control system's response is obtained via simulation. The response and actuator effort are then numerically checked to ensure that various constraints (overshoot, maximum actuator effort, disturbance rejection, etc.) have been met. If these constraints are met, the settling time is then calculated and used as the metric for choosing and evaluating future parameter set guesses. Eventually, this optimization routine will find a set of OLIS and PID parameters which minimize the controlled system's settling time. Unfortunately, because this nonlinear optimization routine attempts to solve for so many variables (usually six or more in the case of OLIS/PID controllers), the minimum found can often be a local minimum, instead of the desired global minimum.

Because of the local minima problem often associated with complicated nonlinear optimizations, the solution routine proposed in this thesis is divided into two distinct parts: a smaller, nonlinear optimization routine and a linear optimization routine. The overall solution routine is depicted in Figure 5.27. The first part uses MATLAB's "fmincon" to choose the PID gains and to check for stability and disturbance rejection constraints. For a given set of PID gains, the second step of the new solution routine will use a Quadratic Programming linear optimization algorithm. This linear optimization will determine the reference input signal that minimizes the settling time measure while ensuring that the overall system meets it overshoot and actuator effort constraints. By solving for as many parameters as possible via a linear optimization technique, the overall solution routine yields better and more consistent results. That is, it is less susceptible to finding local minima.

Another difference between the two solution routines depicted in Figures 5.26 and 5.27 is the form of the reference command given to the OLIS/PID controller. The fully-nonlinear routine (Figure 5.26) solves for the parameters of an input shaper (a small set of impulse times and amplitudes). This input shaper is then used to filter the step reference command used to drive the overall system. On the other hand, the partially-linear optimization routine solves for the reference command ($R(t)$) that will best drive the overall system to follow a step. The reference command solved for by the Quadratic Programming routine is a discrete signal, where the number of discrete points creating the reference signal is determined by the user. The number of discrete points is much higher than the number of input shaper impulses used in the fully-nonlinear optimization routine. Usually, the number of discrete points used for $R(t)$ is over one hundred.

5.2.1.1 Justification for Use of Quadratic Programming

Quadratic Programming is a linear optimization routine that finds the vector $\vec{x}$ which minimizes the function:

$$g(\vec{x}) = \frac{1}{2}\vec{x}^T Q\vec{x} + f^T\vec{x} \tag{5.7}$$

If Q is a positive definite matrix, then the minimization problem will always have a unique solution, $\vec{x}_{min}$.

The first step in the partially-linear optimization routine uses “fmincon” to choose the closed-loop system's PID gains. With these gains established, "fmincon" can then easily check the closed-loop system's stability and disturbance rejection constraint (assuming this constraint is written as a lower limit on any closed-loop pole's time constant). In addition, the closed-loop system can now be written as a linear, state-space system with input r (reference command) and output y. Solving for the reference command that forces the linear, state-space system to follow a desired trajectory can be formulated as a Quadratic Programming problem. The following derivation will show this.

The state-space equations for the closed-loop system are:

$$\dot{x} = Ax + Br \tag{5.8}$$

$$y = Cx \tag{5.9}$$

The solution to these equations at any discrete time step is:

$$x_{i+1} = \Phi x_i + \Gamma r_i \tag{5.10}$$

where,

$$\Phi = e^{AT} \qquad \Gamma = B\int_0^T e^{A\tau} d\tau \tag{5.11}$$

and T is the discrete time step. Next, we define:

$$\beta_i = \left[\Phi^{i-1}\Gamma \;\; \Phi^{i-2}\Gamma \; \ldots \; \Phi\Gamma \;\; \Gamma\right] \tag{5.12}$$

and:

$$\beta^{(n)} = \left[\beta_n \;\; 0^{(\delta,k-n)}\right] \tag{5.13}$$

where $0^{(\delta,k-n)}$ is a matrix of zeros with the row-column dimensions indicated within the parentheses. Here, δ is the number of states (length of the vector x), k is the pre-chosen number of discrete points over which the reference input and system output will be solved, and n is any time step $(0 < n \leq k)$.

If the entire reference signal r is written as a column vector,

$$\vec{R} = [r_0 \ r_1 \ \ldots \ r_{k-2} \ r_{k-1}]^T \tag{5.14}$$

then any state or output value can be written as:

$$x_i = \beta^{(i)} R \qquad y_i = C\beta^{(i)} R \tag{5.15}$$

Likewise, the entire output vector can be written as:

$$\vec{Y} = \left[C\beta^{(1)} \ \ldots \ C\beta^{(k)}\right]^T \vec{R} \ \ or \ \ \vec{Y} = F_y \vec{R} \tag{5.16}$$

where $F_y = \left[C\beta^{(1)} \ \ldots \ C\beta^{(k)}\right]^T$.

A similar process can be completed that will give the actuator input u (which directly acts upon the plant G within the feedback loop) as a function of r:

$$\vec{U} = F_u \vec{R} \tag{5.17}$$

However, the D matrix is usually not zero, as was the case in (5.9). This makes the solution slightly more complicated.

In order to use a Quadratic Programming technique, the entity being minimized has to be expressible via the form shown in (5.7). Therefore, the settling time was minimized by minimizing the sum of the square of the error between the actual system output, y, and the original, desired reference signal (which is a step command in this section). If the original, desired reference is a unit step, then the error signal can be written as:

$$E = \sum_{i=1}^{k} (y_i - 1)^2 = \sum_{i=1}^{k} y_i^2 - \sum_{i=1}^{k} 2y_i + \sum_{i=1}^{k} 1 \tag{5.18}$$

Minimizing this error yields:

$$min(E) = min\left(\sum_{i=1}^{k} y_i^2 - \sum_{i=1}^{k} 2y_i\right) = min\left(\vec{Y}^T * \vec{Y} - 2^{1,k} * \vec{Y}\right) \tag{5.19}$$

Fortunately, the terms within the parentheses of the right-most term can be written as a function of the reference input vector:

$$g(R) = \frac{1}{2}\vec{R}^T Q \vec{R} + f^T \vec{R} \tag{5.20}$$

where Q can be shown to be positive definite and of the form:

$$Q = 2\left[C\beta^1 \ \ ... \ \ C\beta^k\right] * \left[C\beta^1 \ \ ... \ \ C\beta^k\right]^T \tag{5.21}$$

The second term, f, can be written as:

$$f = -2\left[C\beta^1 \ \ ... \ \ C\beta^k\right] \tag{5.22}$$

Now, any standard Quadratic Programming algorithm can be used to minimize $g(R)$ subject to the any number of linear constraints on R. In other words:

$$min(g(R)) \ \ s.t. \ \ A^* \leq b^* \tag{5.23}$$

For the partially-linear optimization routine proposed in this chapter, A^* and b^* are used to enforce the overshoot and actuator limits. For these constraints, A^* and b^* would be of the form:

$$A^* = [F_y \ \ F_u \ \ -F_u]^T \tag{5.24}$$

$$b^* = \left[M_p Limit^{(k,1)} \ \ U_{max}^{(k,1)} \ \ \ U_{max}^{(k,1)}\right]^T \tag{5.25}$$

Note that A^* and b^* can be modified to include any number of linear constraints on R.

5.2.2 Comparing Partially-Linear Optimization to a Standard Nonlinear Optimization Package

One of the big questions with any nonlinear optimization routine is local minima. Because this partially-linear design process uses MATLAB's "fmincon" to find K_p, K_d and K_i, this issue of local minima needs to be addressed. However, this partially-linear optimization routine uses a quadratic programming process to compliment the nonlinear “fmincon” and reduce the number of variables for which it must solve. This is the key difference that allows for more-reliable and superior solutions. To verify this statement, the results obtained from the partially-linear optimization routine are compared to the solutions obtained by a fully-nonlinear optimization procedure that solely utilized MATLAB’s “fmincon” to solve for every variable (the PID gains and the input shaper parameters). These two solution methods are depicted in Figures 5.26 and 5.27.

The following sections will highlight the differences between these two optimization routines by comparing their ability to solve for controller parameters and produce fast step responses. Both design routines were fully executed and solved for PID gains and parameters by which to alter the step-reference command (input shaping parameters or $r(t)$). However, the resulting PID gains are the only solution parameters that will be compared. The comparison of the PID gains is sufficient to establish the parameter consistency of each design routine. The actual controller responses will also be compared to see which one more closely executes the desired step command. Each example will use a different plant type.

5.2.2.1 Mass Plant

For this example, the plant was a simple mass ($m = 1$) and the feedback controller was restricted to the PD form:

$$G = \frac{1}{s^2} \tag{5.26}$$

$$C = K_p + K_d s \tag{5.27}$$

Figures 5.28 and 5.29 show the PD gain solutions for ten randomly initiated iterations of each optimization routine. Remember that optimization routines generally need some starting guess from which to begin searching for the optimal solution. The initial gain guesses are shown on the abscissa of each figure and are denoted by the

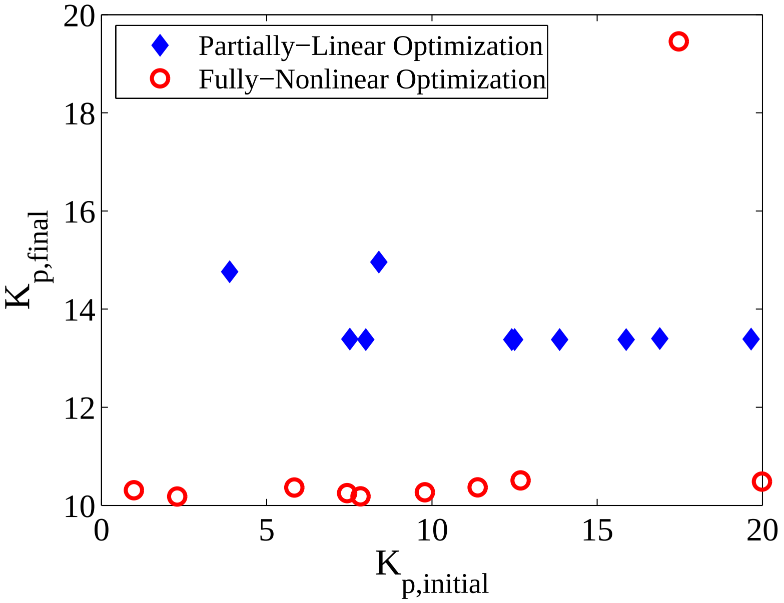


Figure 5.28: K_p Gains Chosen for Mass Plant.

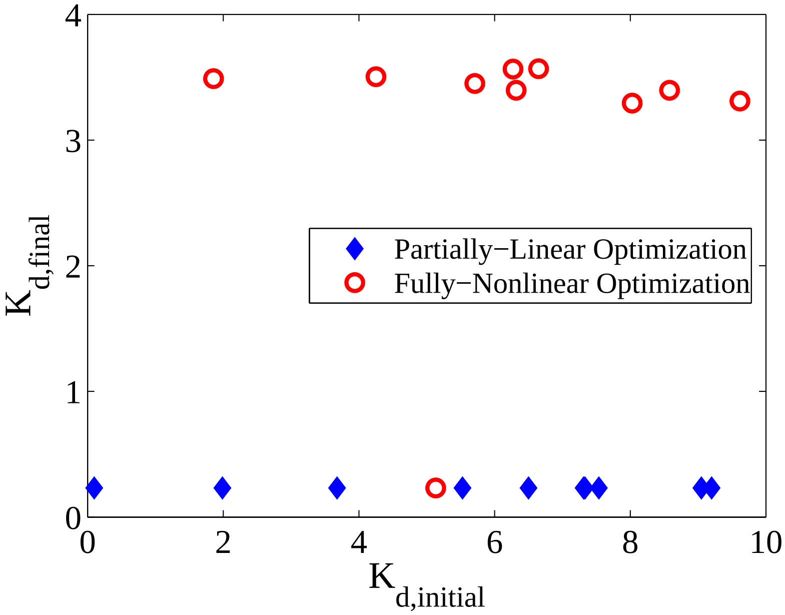


Figure 5.29: K_d Gains Chosen for Mass Plant.

subscript "initial". The gains eventually chosen as "optimal" by the respective optimization techniques are shown on the ordinate of each figure. These final values are denoted by the "final" subscript.

If the outliers in Figure 5.28 (those $K_{p,final}$ values above 14) are neglected, then

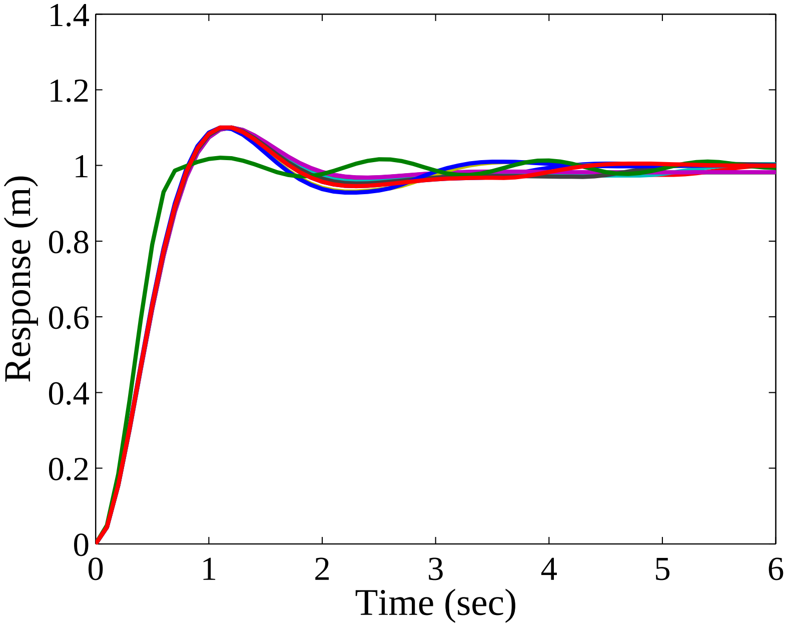


Figure 5.30: Comparing Fully-Nonlinear Optimization Solutions for a Mass Plant.

the partially-linear optimization routine returns a more consistent set of K_p gains than did the fully-nonlinear optimization routine. This is also true for the K_d gains, without having to disregard any outliers.

Another way to compare these two optimization routines is to graphically compare the responses of the controlled systems designed by each routine. Figure 5.30 shows the ten responses to each of the ten controlled systems designed by the fully-nonlinear optimization procedure. Figure 5.31 shows the ten responses for each of the ten controlled systems solved for by the partially-linear optimization routine. For this mass plant, the partially-linear optimization routine produced a more consistent set of controllers than did the fully-nonlinear routine. In addition, the ten responses produced by each of the ten controllers designed by the partially-linear optimization routine were superior (in terms of settling time) than the responses from the fully-nonlinear optimization routine. Remember that settling time was the parameter being minimized.

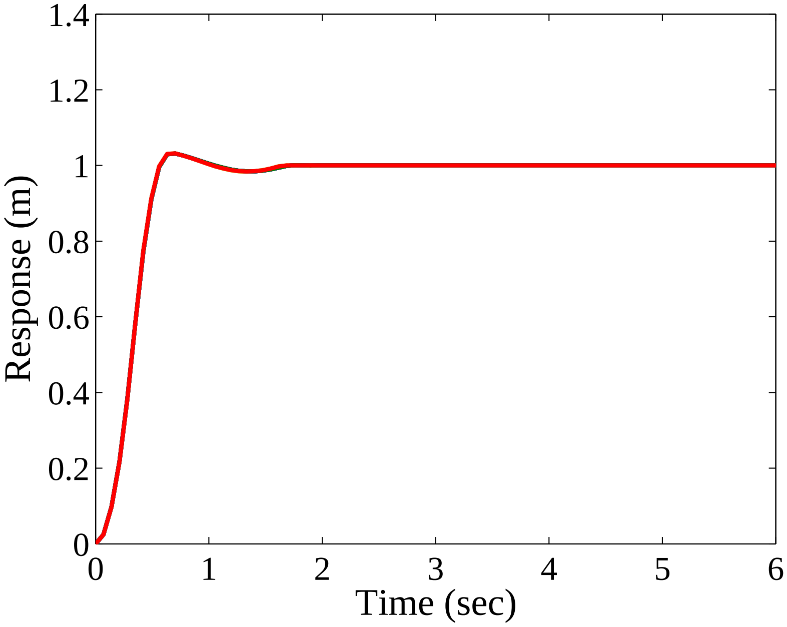


Figure 5.31: Comparing Partially-Linear Optimization Solutions for a Mass.

5.2.2.2 Damped, Second-Order Plant

For this example, the plant was a damped, second-order system and the feedback controller was of the PID form:

$$G = \frac{9}{s^2 + 0.6s + 9} \tag{5.28}$$

$$C = K_p + K_d s + \frac{K_i}{s} \tag{5.29}$$

The results are depicted in a manner similar to the previous section. Figures 5.32 - 5.34 show the PID gains chosen by each optimization routine as a function of the starting guesses. Again, the K_p and K_d gains are very consistent when chosen by the partially-linear optimization. However, the K_i gain solutions have a consistency roughly equal to those obtained by the fully-nonlinear solution.

Despite the uncertainty in the integral gain solution, the partially-linear optimization scheme still yielded consistent and fast overall control schemes. Figure 5.35 shows the slight inconsistency of the responses to the controlled systems obtained from the

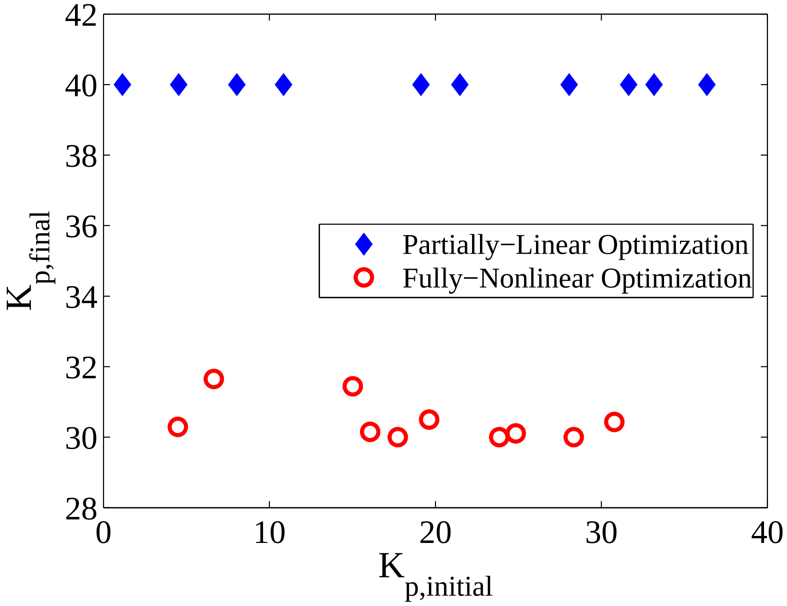


Figure 5.32: K_p Gains Chosen for Second-Order Plant.

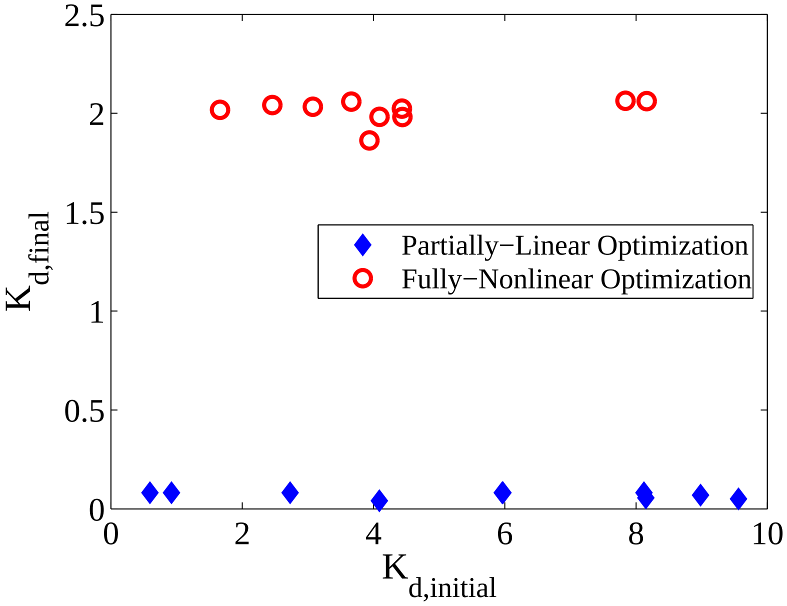


Figure 5.33: K_d Gains Chosen for Second-Order Plant.

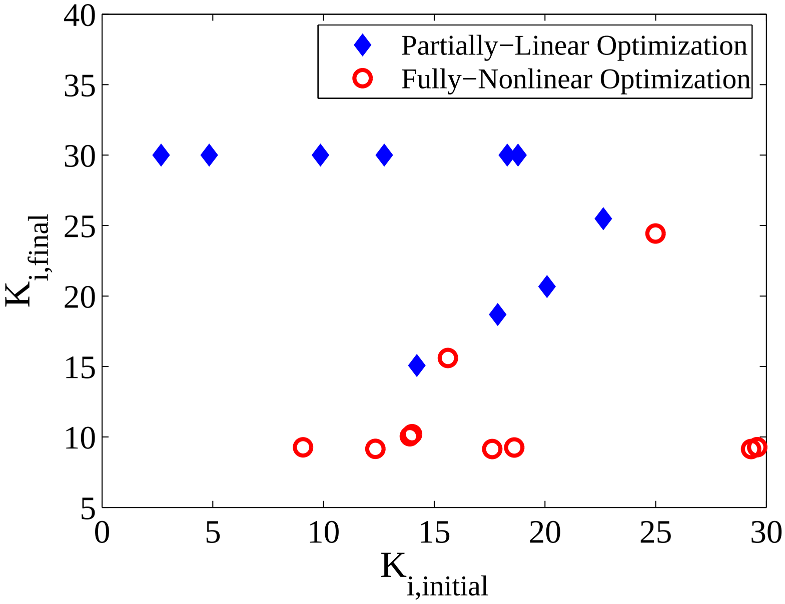


Figure 5.34: K_i Gains Chosen for Second-Order Plant.

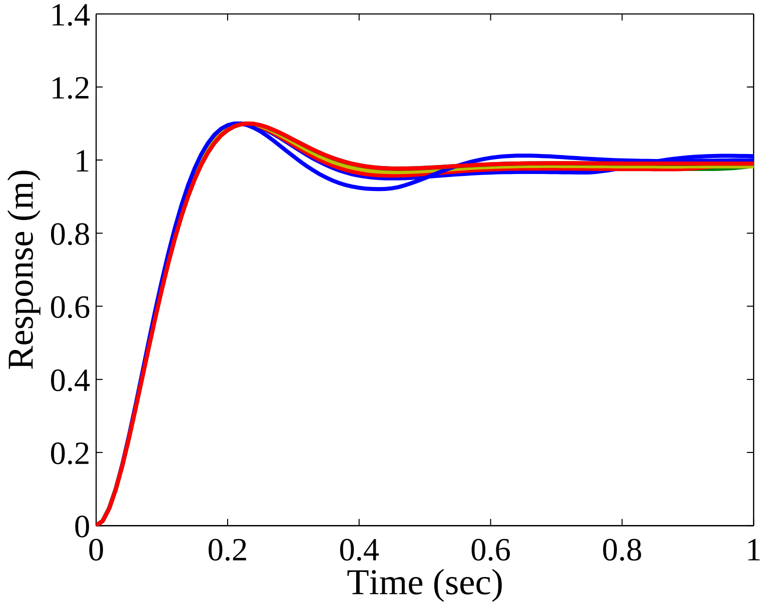


Figure 5.35: Fully-Nonlinear Optimization Solutions for a Second-Order Plant.

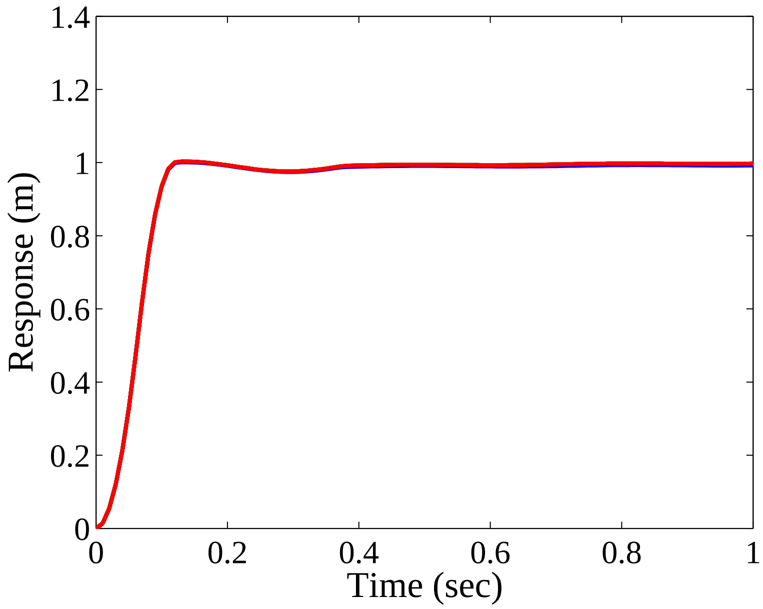


Figure 5.36: Partially-Linear Optimization Solutions for a Second-Order Plant.

fully-nonlinear optimization routine. Figure 5.36 shows the ten step responses to the controlled systems solved for by the partially-linear routine. Again, this optimization routine was able to achieve controlled systems with faster step responses, less overshoot and more consistency.

5.2.2.3 Third-Order Plant with Numerator Dynamics

For this example, the plant was a third-order system with one zero and the feedback controller was of the PID form:

$$G = \frac{s+a}{s^3 + bs^2 + cs + d} \tag{5.30}$$

$$C = K_p + K_d s + \frac{K_i}{s} \tag{5.31}$$

Here, $a = 1.5$, $b = 2$, $c = 3.2$ and $d = 1.5$.

Figures 5.37 - 5.39 show the results obtained by both optimization routines. Notice that for this plant, the resulting PID gains have approximately the same degree of consistency. However, the actual responses to each controlled system show the same pattern as the two previous examples. The responses for the fully-nonlinear routine

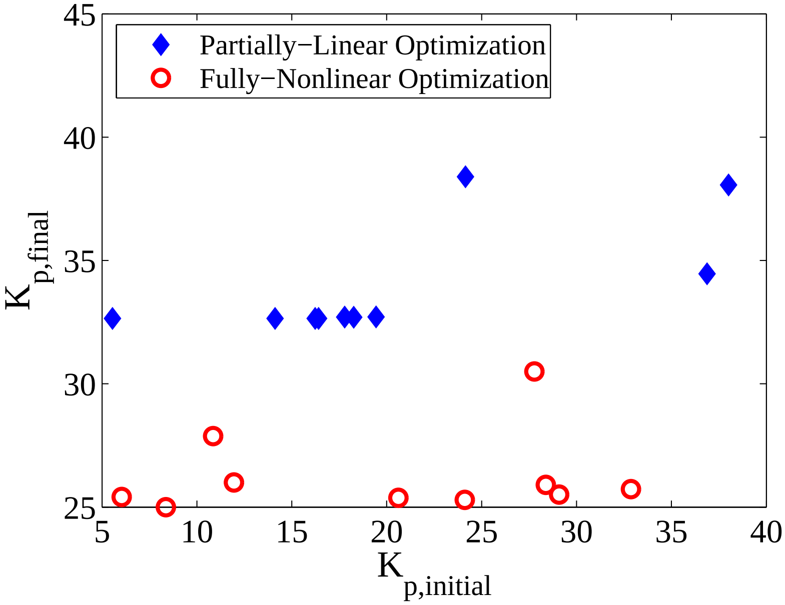


Figure 5.37: K_p Gains Chosen for Third-Order Plant.

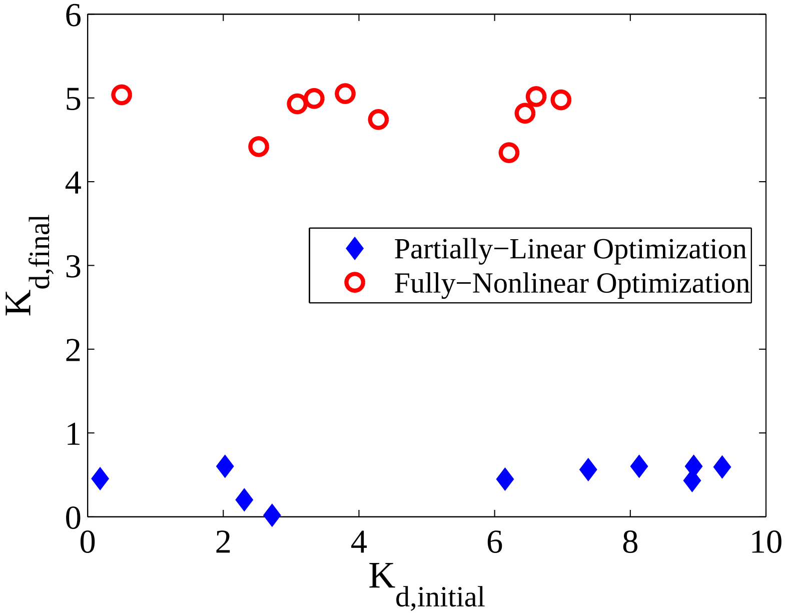


Figure 5.38: K_d Gains Chosen for Third-Order Plant.

are inconsistent, as shown in Figure 5.40. The responses for the partially-linear optimization routine shown in Figure 5.41 are again more consistent, faster, and have less overshoot.

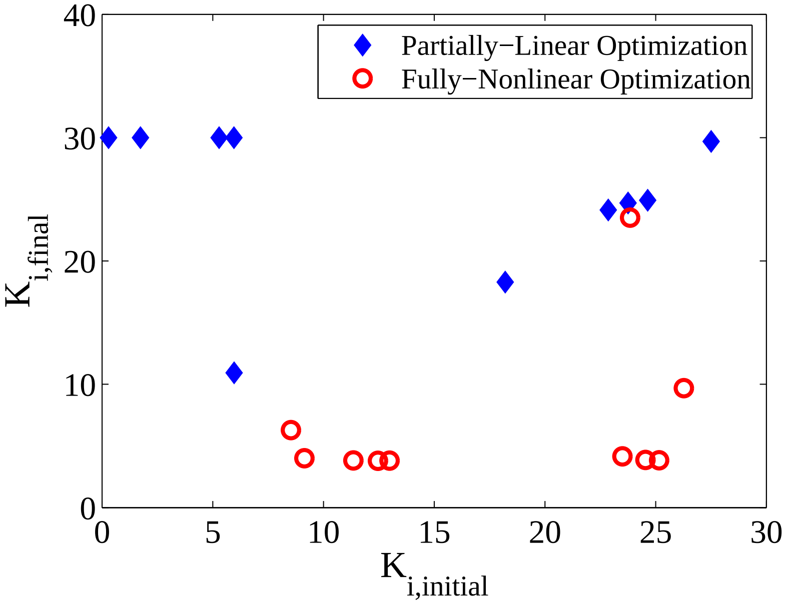


Figure 5.39: K_i Gains Chosen for Third-Order Plant.

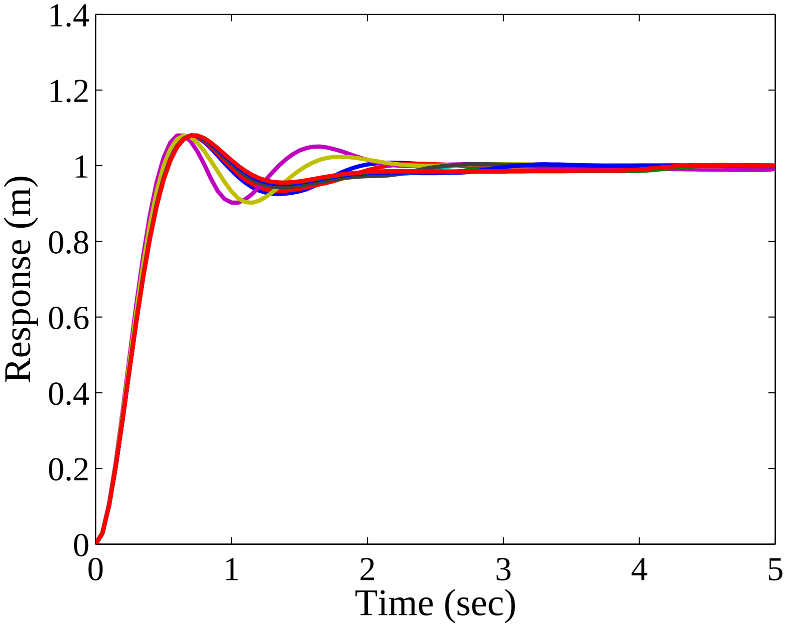


Figure 5.40: Fully-Nonlinear Optimization Solutions for a Third-Order Plant.

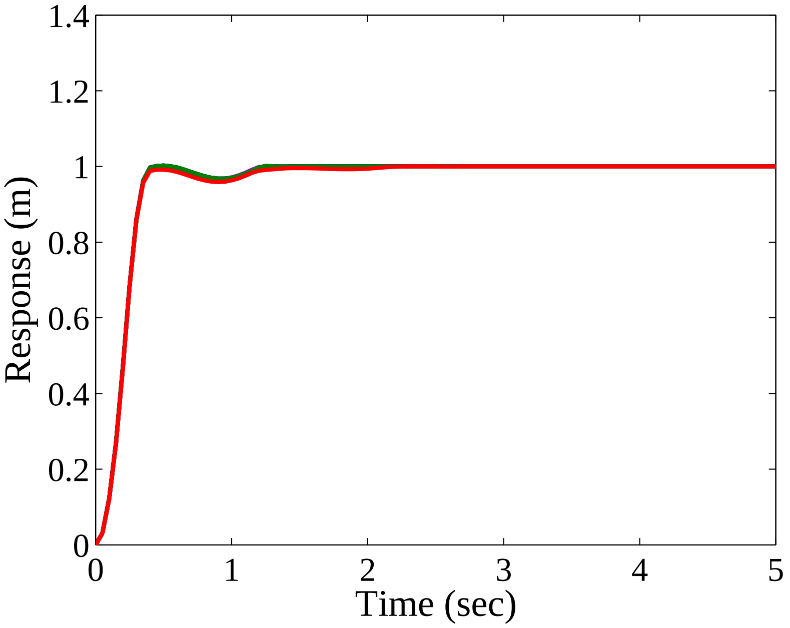


Figure 5.41: Partially-Linear Optimization Solutions for a Third-Order Plant.

5.2.2.4 Fourth-Order Plant with Numerator Dynamics

The final plant type studied here is a fourth-order plant (mass-spring-mass-spring-ground where the input, sensor and output are all located at the first mass) again under PID control:

$$G = \frac{\alpha s^2 + \beta}{s^4 + \delta s^2 + \gamma} \tag{5.32}$$

$$C = K_p + K_d s + \frac{K_i}{s} \tag{5.33}$$

The plant coefficients depend on the masses and spring constants. Here, $\alpha = 0.333$, $\beta = 0.4267$, $\delta = 1.68$, $\gamma = 0.32$.

Testing ten different PID gain initial guesses, the results of both optimization schemes are shown in Figures 5.42 - 5.44. The first thing to note is that both solution routines occasionally failed to find a reasonable solution (less than ten final values are shown in Figures 5.42 - 5.44). In these cases, “fmincon” hit its internal iteration limit. This limit can certainly be adjusted, but it was found that increasing this limit too much resulted in the partially-linear optimization sometimes requiring an extremely

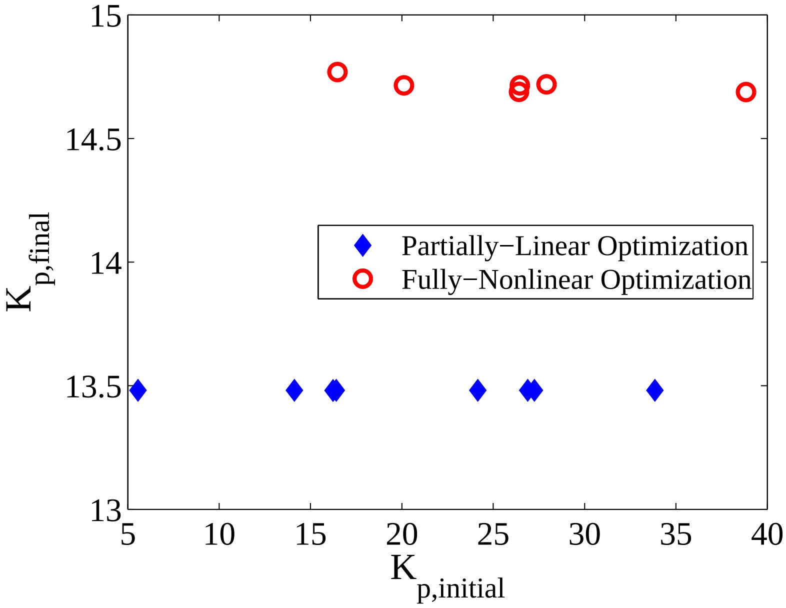


Figure 5.42: K_p Gains Chosen for Fourth-Order Plant.

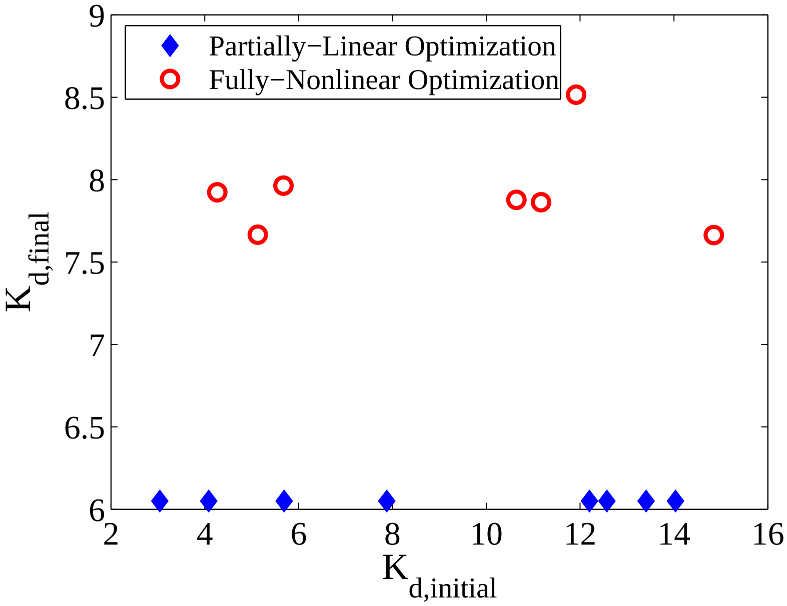


Figure 5.43: K_d Gains Chosen for Fourth-Order Plant.

long amount of time to find a solution (often an order of magnitude longer in time than before this "fmincon" limit was raised).

However, for the cases when the partially-linear optimization does find a solution, the optimal PID gains are very consistent. In fact, they are much more consistent

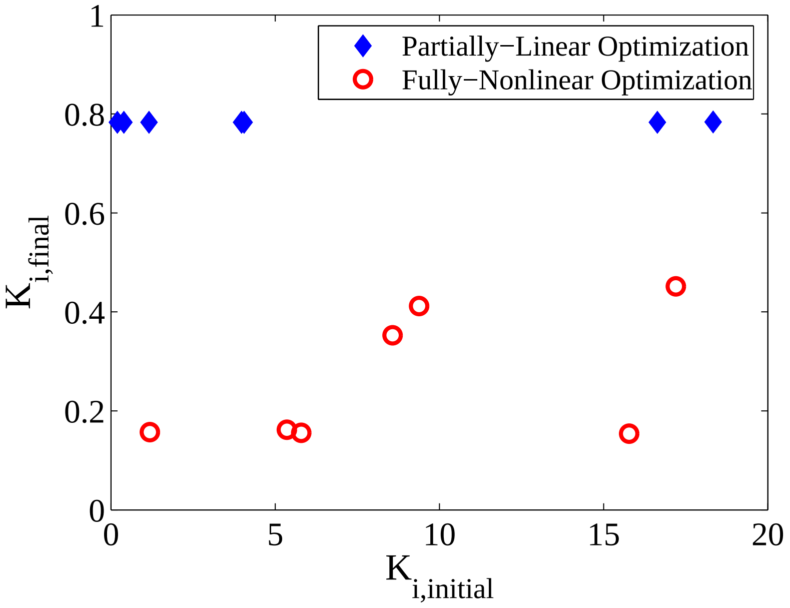


Figure 5.44: K_i Gains Chosen for Fourth-Order Plant.

than the fully-nonlinear optimization solutions. The standard deviations for each of the PID gain solution sets found by the partially-linear optimization routine were between 0.0002 and 0.0003. For the fully-nonlinear optimization routine, the standard deviations for each of the PID gain solution sets were between 0.1152 and 0.286.

The responses to each of the controlled systems successfully designed by both optimization routines are shown in Figures 5.45 and 5.46. The partially-linear optimization routine again yields more consistent and superior system responses.

5.2.2.5 Comparing Concurrent and Sequential Design

Finally, it can be shown that the partially-linear, concurrent design optimization technique proposed in this thesis can outperform a similar-structured sequential design technique. Here, the sequential design technique first runs the outer nonlinear minimization routine ("fmincon") that chooses the PID gains by minimizing the error (from desired position) assuming the reference input is always a unit step. Then, with the chosen PID gains, the quadratic programming routine is called to design a non-step reference input to further minimize the error (from desired position), as well

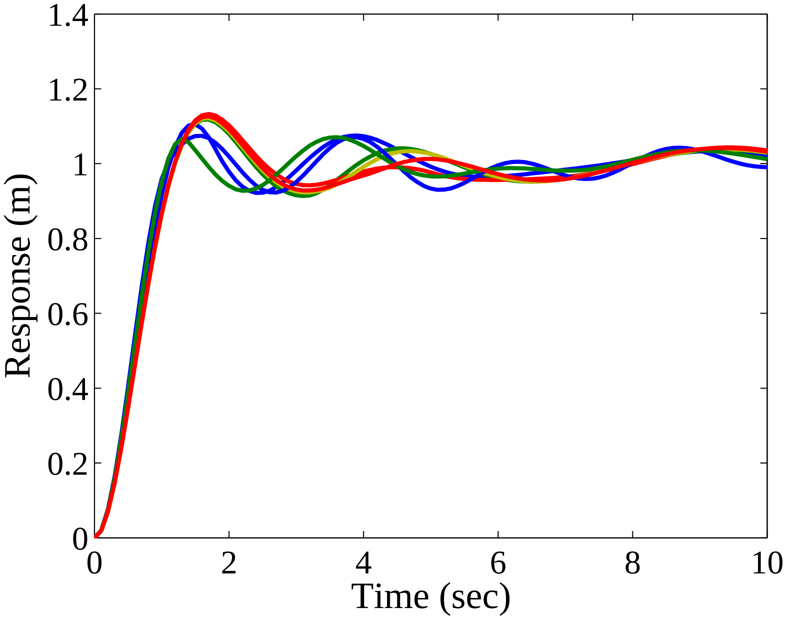


Figure 5.45: Fully-Nonlinear Optimization Solutions for a Fourth-Order Plant.

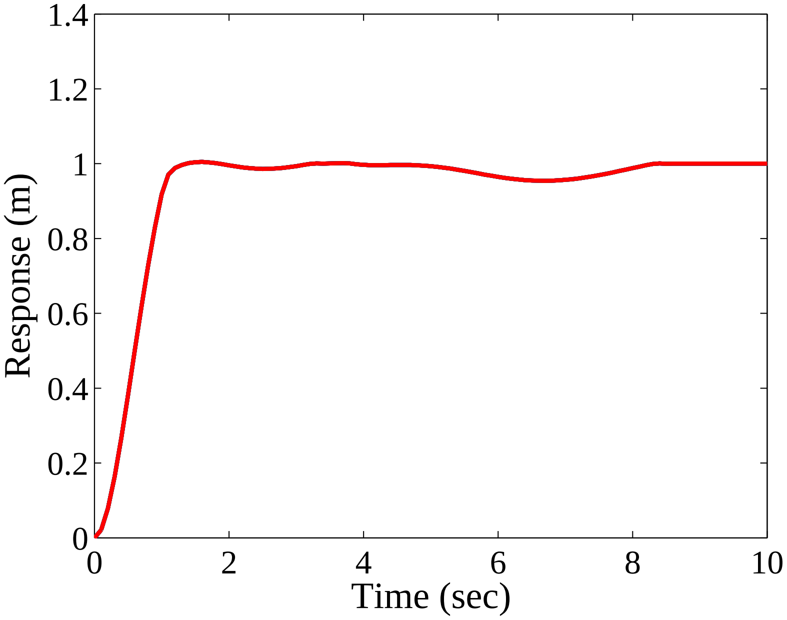


Figure 5.46: Partially-Linear Optimization Solutions for a Fourth-Order Plant.

as to meet constraints (overshoot, maximum actuator effort and disturbance rejection limits). Figure 5.47 compares the two step responses, showing that the concurrent design is faster than the sequential design.

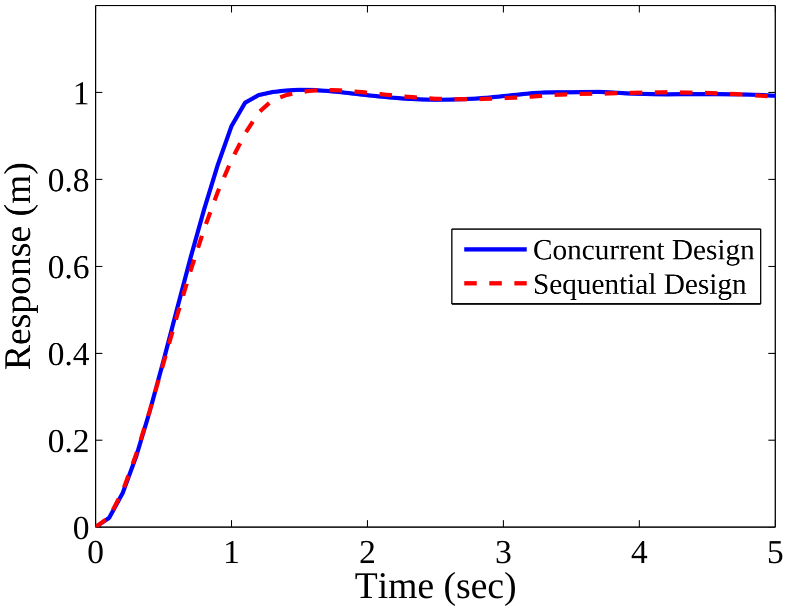


Figure 5.47: Concurrent vs. Sequential Design on a Fourth-Order Plant.

5.2.2.6 Current Deficiencies in the Partially-Linear Optimization Scheme

There are a few problems associated with the newly proposed partially-linear optimization scheme. First, Section 5.2.2.4 demonstrated how the partially-linear optimization routine can sometimes fail to find a solution. Two of the ten solution iterations failed to find a viable solution. This problem can be exasperated by forcing strict design constraints that reduce the solution space. For example, the same optimization that was carried out in Section 5.2.2.4 was repeated with a stricter disturbance rejection constraint. By adjusting this constraint, fewer combinations of PID gains yielded acceptable design solutions. In fact, only four iterations found a viable solution. By restricting the solution space, the partially-linear optimization routine was more likely to fail than before. However, it should be noted from the results in Section 5.2.2.4 that the fully-nonlinear optimization also fails on occasion. It is unknown if one of the optimization routines is inherently superior to the other in this regard.

This solution-finding dependance upon performance constraints can also be viewed as a method to improve an optimization routine's ability to find solutions. If certain constraints can be relaxed, then this will allow more PID combinations to result in viable solutions. This will generally make it easier for an optimization routine to search through the solution space and find a minimum.

Obviously, one solution to this solution-finding problem is to adjust the MATLAB parameters that set internal, iteration limits. However, as briefly noted before, this often led to the partially-linear optimization running for extremely long times (sometimes well over one hour). Normal solution times typically ranged from 5 to 20 minutes. Note that raising the iteration limit can also cause the fully-nonlinear optimization routine to take a significant amount of time.

The two previously described limitations indicate that there is a functional limit as to the systems, or scenarios, that this newly proposed optimization technique can reasonably address. If the system has a very constrained solution space, then this new optimization routine will have a higher failure rate. Also, if the plant being studied combines slow and fast dynamics, then any response simulation will need to be carried out in small time steps (to accurately simulate the fast dynamics) and for a long period of time (to accurately simulate the slow dynamics). This will force the quadratic programming function to require significant computation time, further increasing the overall optimization's run time.

The last major problem with the newly proposed partially-linear optimization scheme is the form of some of the reference inputs returned as solutions. Figure 5.48 shows one of the reference inputs required for the response solutions shown in Figure 5.46. The reference input never settles, periodically deviating from unity to suppress a very lightly-damped mode of the closed-loop system. One solution to this problem is to force the closed-loop system to have more damping (for example, by increasing the disturbance rejection constraint). The low-vibration, second mode could then be allowed to damp out naturally.

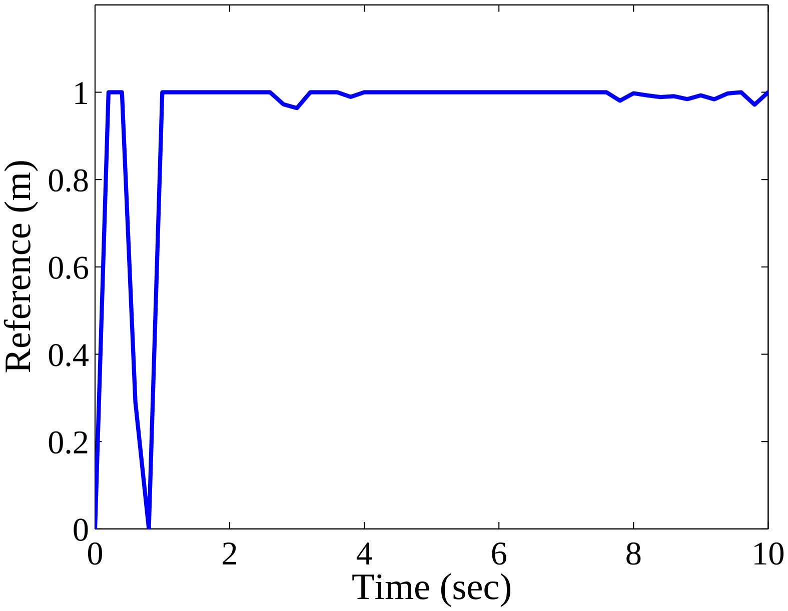


Figure 5.48: Reference Input from the Partially-Linear Optimization.

However, in the case that the closed-loop system will have a lightly-damped, secondary-mode, another solution is possible. The first step is to force the quadratic programming routine to find a reference input that stays at unity past a certain time. The second step is to only require the optimization routine to minimize the system error after a certain time. This step is added in recognition that no system can perfectly track a step command. Some error will always exists as the system moves from its current position to its desired position.

Figure 5.49 shows ten partially-linear optimization routine runs (using the same fourth-order plant) with the new constraints presented in the previous paragraph. In addition, the reference input obtained by the quadratic programming routine was forced to find an input that was monotonically increasing. This was added to avoid reference commands with such sharp transitions like the one shown in Figure 5.48. These quickly-changing reference commands are often desired, as they cause systems to move quickly. However, they can also be too demanding for some actuators, and they can excite higher modes of vibrations.

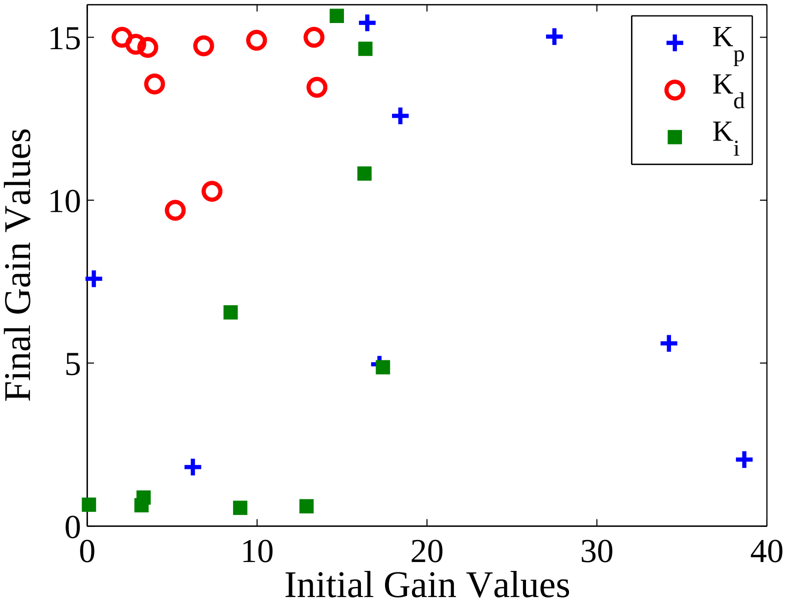


Figure 5.49: PID Gains Chosen Under Redesigned Error Minimization.

As shown in Figure 5.49, the solutions are definitely not consistent. This is to be expected in this case, because the system error was only minimized after a certain time. Before that time, the system is allowed to have any response. Therefore, it makes sense that multiple solutions would be obtained. The responses to these solutions can be seen in Figure 5.50. The reference inputs used to create these responses are shown in Figure 5.51. Because multiple solutions are bound to occur in this form of the partially-linear optimization routine, it would be preferable to run the routine multiple times and pick the visually-best response. For example, the reference input and response shown in Figure 5.52 constitutes one of the fastest responses with zero overshoot shown in Figure 5.50.

5.3 Discussion: Concurrent Design of OLIS/PID Controllers for Generic Plants

The first half of this chapter developed new rules of thumb for combining outside-the-loop input shaping and PID feedback control. First, a time domain perspective was taken. This resulted in new tuning rules specifically designed for OLIS/PID

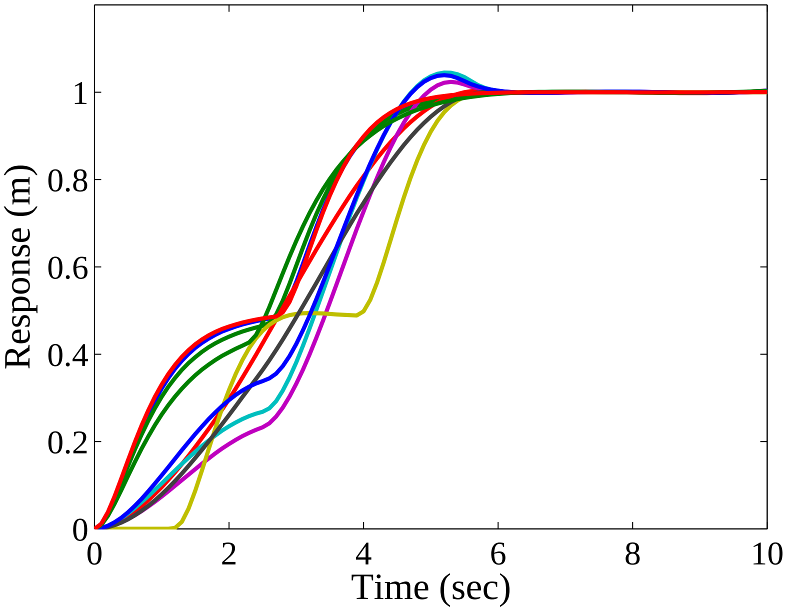


Figure 5.50: Step Responses Under Redesigned Error Minimization.

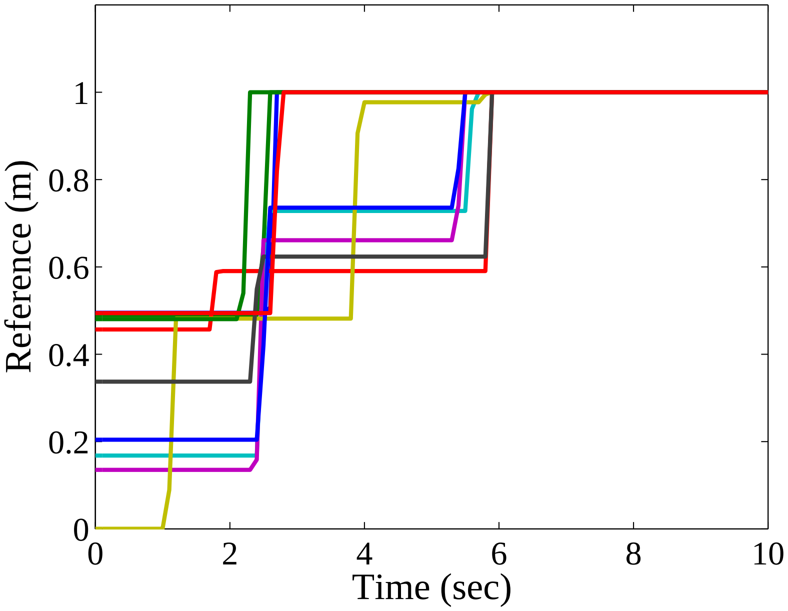


Figure 5.51: Reference Inputs for Redesigned Error Minimization.

controllers. Secondly, a frequency domain perspective was utilized to choose PID gains via the Bode diagram. In both cases, the knowledge of the input shaper allowed for the design of more aggressive close-loop systems. The final result was a faster step response that still met various design constraints.

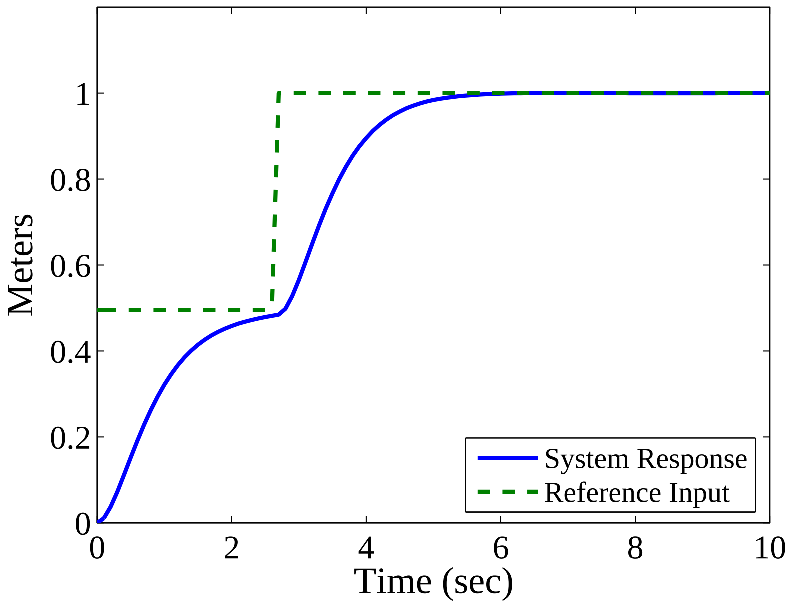


Figure 5.52: Best Solution Under Redesigned Error Minimization.

The second half of this chapter looked at precise, numerical optimization routines for choosing PID gains and input shaper parameters (or the entire reference input). The newly proposed optimization routine relied (as much as possible) on a Quadratic Programming routine. Only those parameters which could not be solved by the linear optimization routine were solved via a nonlinear method. The controlled systems generated by this partially-linear optimization routine were compared to systems generated by a fully-nonlinear optimization routine. This fully-nonlinear optimization routine solved for all system parameters and checked all constraints via a standard nonlinear optimization package know as "fmincon". The comparison indicated that the partially-linear optimization routine generally chose a more consistent set of PID gains. Although, this was not always the case. However, the actual step responses to the overall systems designed by the partially-linear optimization routine were always more consistent and superior to those obtained from the systems designed by the fully-nonlinear optimization routine. By utilizing linear optimization, the partially-linear optimization routine was better at avoiding local minima and finding superior solutions.

It was also shown in the previous section that the partially-linear optimization routine has several drawbacks. As can happen in any optimization, the routine can sometimes fail to find a solution or require a significantly long solution time.

The use of complex, optimization routines is often necessary. Effects such as secondary modes and numerator dynamics (including non-minimum phase systems) will often make the equations governing pure second-order systems impractical to use. Fortunately, numerical optimization techniques based upon simulating the system's response can handle these issues and address the vast majority of plant types.

CHAPTER VI

STABILITY ANALYSIS OF CLOSED-LOOP SIGNAL SHAPING CONTROLLERS

Having extensively studied the intelligent combination of outside-the-loop input shaping and PID feedback control, this dissertation will now focus on feedback controllers that contain an input shaping filter somewhere within the feedback loop. These controllers are called closed-loop signal shaping (CLSS) controllers. The basic assumptions associated with this dissertations study of CLSS are:

1. G is linear, time-invariant
2. G is minimum phase
3. The closed-loop system is a single-input, single-output (SISO) system

Each particular study within the following four chapters will add additional assumptions that are detailed within their respective section. Future work could extend this research beyond the constraints set by assumptions #1 - #3.

Arguably the most important issue when designing control systems is stability. A controller is useless unless it can produce a stable, closed-loop system. Therefore, this chapter will begin the study of closed-loop signal shaping (CLSS) controllers by investigating whether or not they can yield stable behavior. This investigation will use continuous domain root loci and Bode plots, in addition to discrete domain root loci, to understand what drives CLSS controllers unstable and what increases stability margins.

6.1 Introduction to Stability of CLSS Controllers

The Closed-Loop Signal Shaping control scheme considered here is the Classical Method shown in Figure 6.1. Here, "I" is an input shaper and "C" is some other controller. This control scheme is probably the most obvious form of Closed-Loop Signal Shaping, and it is the form most often found in literature. However, as with any form of CLSS, this control scheme presents an obvious question of stability. Input shapers utilize time delays. Within feedback loops, time delays are known to cause stability problems. Therefore, in order for the Classical Method to be a reasonable approach, it must be established that closed-loop stability is possible.

As discussed in Chapter 1, a few noteworthy stability studies have previously been presented. However, what is missing from the literature is a detailed stability investigation of the Classical Method that analyzes the control scheme with basic, classical stability tools. Therefore, this chapter analyzes the Classical Method via a root locus and Bode perspective. Within this study, the effects of the most important system parameters (damping, frequency, etc.) are analyzed. The result is a more general and intuitive understanding of the stability issues surrounding the Classical Method. In addition, the analysis facilitates the design of stable CLSS controllers, even in the midst of parameter uncertainties.

Section 6.2 will describe the stability of a second-order system within the continuous domain. In Section 6.3, the stability of a second-order system will be analyzed in the digital domain. Section 6.4 will extend the continuous domain discussion by

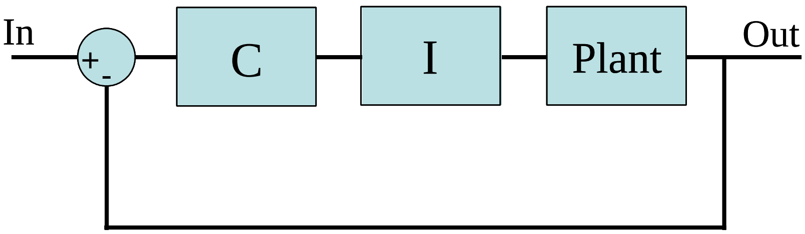


Figure 6.1: Block Diagram of the Classical Controller Method.

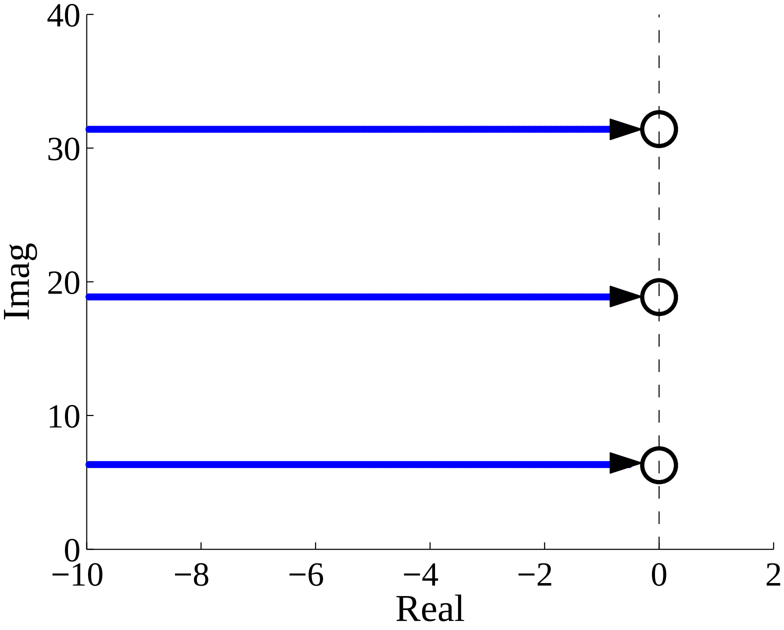


Figure 6.2: Root Locus of the Simplified, CLSS Controller.

analyzing a fourth-order plant with numerator dynamics. Finally, Section 6.5 will present some experimental results on the stability of CLSS controllers.

6.2 Stability Analysis of a Second-Order Plant

6.2.1 Initial Insights

To begin the stability study, the controller, "C", in Figure 6.1 was restricted to be a proportional gain, K. If the plant is assumed to be unity, then the closed-loop system is only a function of the input shaper and the proportional gain (K). The root locus of the system, when $C = K$ and $Plant = 1$, is shown in Figure 6.2. Chapter 1 discussed how input shapers have an infinite column of open-loop zeros and an infinite number of open-loop poles located at $s = -\infty \pm j\omega$. The root locus plot in Figure 6.2 shows that the infinite number of open-loop poles create an infinite number of root locus branches and closed-loop poles when a shaper is included within the feedback loop. Note that the closed-loop poles arising from the input shaper tend to form a column that extends infinitely away from the real axis. This occurs because of the infinite

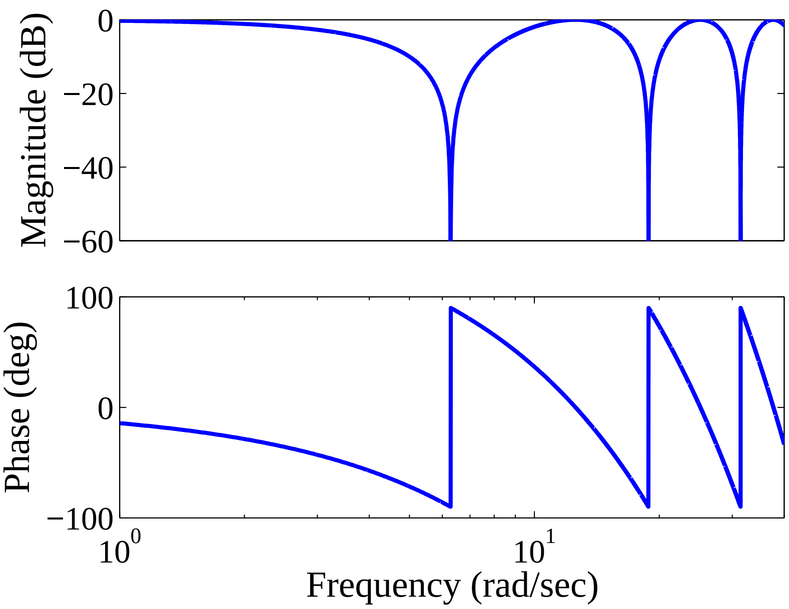


Figure 6.3: Open-Loop Bode Diagram of a ZV Shaper.

column of open-loop zeros, to which the closed-loop poles approach. However, in the general case, the the input shaper's open-loop zeros do not lie on the imaginary axis. They can lie anywhere in the left-half plane, depending upon the oscillatory poles they are designed to cancel.

Figure 6.3 shows the Bode diagram of the system considered above. Note that this Bode diagram is simply the frequency response of the input shaper. The discontinuity in the phase plot of the input shaper is a consequence of the Nyquist plot passing through the origin. This also explains why the magnitude plot goes to $-\infty dB$ at these phase discontinuities. These phenomenon occur because the input shaper used here has zeros that are located exactly on the imaginary axis, as shown in Figure 6.2. The Nyquist path, as it moves along the imaginary axis, passes through these zeros. A vector of magnitude zero has an undefined angle, hence the phase discontinuities. If the input shaper zeros lie in the left half plane, then the phase discontinuities in the Bode plot become continuous, but sharp, transitions from $\sim -90°$ to $\sim +90°$.

The root locus and Bode plots previously shown constitute the first insight into

CLSS controllers. Using input shapers within feedback loops will add an infinite number of root locus branches. The input shaper will also severely attenuate certain frequencies and will both add and subtract phase throughout the entire frequency spectrum.

The second insight concerns the basic effect that modeling errors will have on closed-loop stability. The plant, “G”, is now chosen to be an undamped second-order system with a natural frequency of ω_a. The modeling error studied here comes from an error in the estimation of the plant's natural frequency. That is, the frequency used to design the input shaper (ω_m) does not equal the actual natural frequency of the undamped second-order system (ω_a). Under such modeling errors, closed-loop stability depends heavily on the relationship between ω_a and ω_m. The main reason for this can be seen in Figure 6.4, where several root locus plots are shown for various values of ω_a. Note that the imaginary portion of the roots has been normalized by ω_m, which is $2\pi\frac{rad}{s}$ in this analysis. The input shaper will add an infinite number of

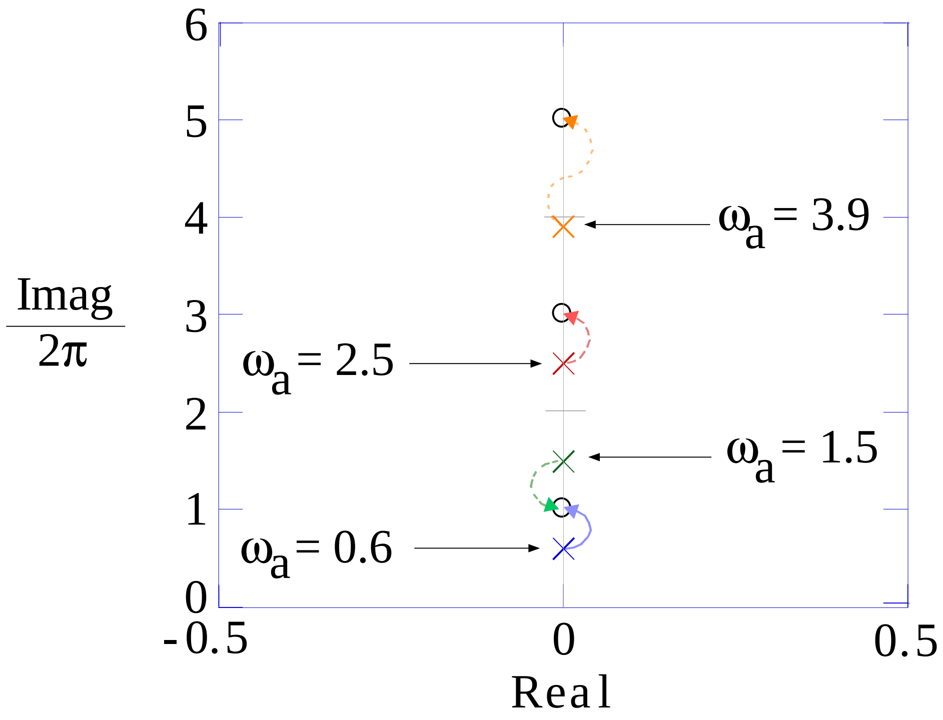


Figure 6.4: CLSS Root Locus Sketches.

zeros along the imaginary axis located at odd multiples of ω_m. These zeros remain stationary in this analysis, but the poles from the plant (located along the imaginary axis at $\pm\omega_a$) move as the plant's natural frequency changes. Root locus theory states that a branch of the root locus will go from each plant pole to some open-loop zero; usually the zero closest to that pole (the branch may extend to ∞ if there are more open-loop poles than open-loop zeros). However, in this analysis, whether the plant pole is above, or below, the zero to which the plant pole is closest, determines whether or not the root locus branch will bend into the right-half plane (RHP) or the left-half plane (LHP). Figure 6.4 shows this trend by depicting 4 different root locus drawings for four different plants (Each color and line style corresponds to a different value of the plant's natural frequency). Depending upon the location of the plant poles relative to the shaper zeros, the root locus from each plant pole goes to different shaper zeros and extends into different halves of the real/imaginary plane.

Usually, if the plant pole is below the shaper zero to which it is closest, the root locus branch extending from that plant pole bends to the right, making the system unstable. On the other hand, if the pole is above the zero to which it is closest, the root locus branch bends left, creating a stable system. When the pole lies close to the middle point between two zeros, its root locus branch can enter both the RHP and the LHP. This case is shown in Figure 6.4 for the plant natural frequency of $\omega_a = 3.9$. Notice how this plant pole is almost equidistant from the two closest input shaper zeros. However, since the plant pole is actually closer to the lower shaper zero, the root locus branch starts begins in the LHP. Finally, it should be noted that the data shown in Figure 6.4 are not actual root loci that were mathematically determined. The plots were simply sketched to explain the phenomena observed in simulation studies. Future sections will show actual root locus plots which corroborate the sketches shown here.

6.2.1.1 Closed-Loop Signal Shaping: Root Loci Proof

Constructing continuous domain root loci of CLSS controllers presents a few interesting challenges. First, the closed-loop characteristic equation is not a polynomial in “s”. This means that numerical methods must be employed to construct the root locus. In fact, this dissertation will use a numeric root locus drawing technique created by Nishioka [54]. In addition, the input shaper contains an infinite column of open-loop zeros, resulting in an infinite number of closed-loop poles. These poles are not necessarily in a straight column, although they do extend vertically away from the real axis as described in Section 6.2.1 and Figure 6.2. This ambiguity as to the location of an infinite number of closed-loop poles can make the use of the continuous domain root locus tool difficult, as it is not obvious what axes range should be drawn to include all significant closed-loop poles. However, it can be shown that beyond a sufficiently large radial distance from the origin, the closed-loop poles arising from the input shaper move left as they get farther from the real axis. This means that the most significant closed-loop poles arising from an input shaper are those closest to the real axis and that a controls engineer can reasonably ignore all but a finite number of these closed-loop poles.

To show this, a simple proof is presented. The characteristic equation of the Classical Method depicted in Figure 6.1 can be shown to be of the form:

$$1 + K * I * F = 0 \tag{6.1}$$

where $K*I*F$ is the open-loop transfer function of the closed-loop system in question. Furthermore, K is a proportional gain, I is the input shaper, and F is the portion of the open-loop transfer function that can be written as a rational transfer function (a fraction of s polynomials). One further assumption is that the denominator of F is of higher order than the numerator.

For some K value, there is an infinite number of s values that will satisfy the characteristic equation. In addition, the closed-loop poles arising from the input shaper dynamics tend to form a vertical column as they approach the column of open-loop zeros established by the input shaper. By moving away from the real axis, the magnitude of these s values tends towards infinity. That is,

$$|s| \rightarrow \infty \tag{6.2}$$

As $|s| \rightarrow \infty$, it follows from the assumption on F that $|F(s)| \rightarrow 0$. However, since $K * I * F$ (with K being a finite constant) must always equal -1, it follows that $|I| \rightarrow \infty$ as $|s| \rightarrow \infty$. The input shaper, I, will have the form:

$$I = A_1 + \sum_{i=2}^{n} A_i e^{-st_i} \tag{6.3}$$

where A_i and t_i are the magnitude and time, respectively, of the i^{th} impulse. Note that every input shaper can be written in this form, as they are all a summation of time-delayed impulses (t_1 is set to zero, following standard practice). Knowing that $s = \sigma + j\omega$, this equation can be rewritten as:

$$I = A_1 + \sum_{i=2}^{n} A_i e^{-\sigma t_i} e^{-j\omega t_i} \tag{6.4}$$

In order for this expression to approach infinity as $|s|$ tends towards infinity, some term in this equation must approach infinity along with $|s|$. The A_i terms are constant and usually less than or equal to one. The $e^{-j\omega t_i}$ term has a constant unity magnitude. Therefore, the $e^{-\sigma t_i}$ term must be the term whose magnitude approaches infinity as $|s|$ does. Because t_i is fixed and finite, σ must approach $-\infty$ as $|s| \rightarrow \infty$. This means that as the closed-loop poles for a single K value lie farther from the real axis, they also lie farther to the left of the imaginary axis. Therefore, at some point, they become insignificant and only a finite portion of the real-imaginary plane is needed for sufficient system identification and control. It should be noted here that the exact relationship defining how the poles for a single K value tend towards the left as they lie farther from the real axis is currently unknown.

One point of caution is the fact that the trend established above is only valid as $|s|$ approaches infinity. There is currently no established method for determining the exact s value for which the trend begins. For small s values, where the closed-loop poles from the input shaper are close to the dynamics arising from F, no specific trend exists. The controls engineer must establish the suitable real-imaginary plane area outside of which the above mentioned trend holds and where the closed-loop poles are insignificant.

6.2.2 Full Stability Analysis of a Damped 2^{nd}-Order System

Given the insights gained by the previous examples, we can now consider a more complicated system. Here, a damped second-order plant of the following form is studied:

$$G = \frac{\omega_a^2}{s^2 + 2\zeta\omega_a s + \omega_a^2} \tag{6.5}$$

To begin, the undamped version is analyzed with root locus and Bode plot tools, then other features like damping and lead compensators are added.

6.2.2.1 Root Locus Analysis

Beginning with the control system depicted in Figure 6.1, the controller, C, is first defined as a proportional controller ($C = K$). If the shaper is exactly tuned to the plant frequency, then the root locus will be similar to that shown in Figure 6.5. Here, the plant parameters are $\omega_a = 2\pi$ and $\zeta = 0$.

As shown in Figure 6.2 and reiterated in Figure 6.5, placing an input shaper within the loop will result in an infinite number of closed-loop poles. Their presence indicates another important result. Using an input shaper within a feedback loop, as done within the control scheme depicted in Figure 6.1, will result in additional oscillatory dynamics arising solely from the input shaper.

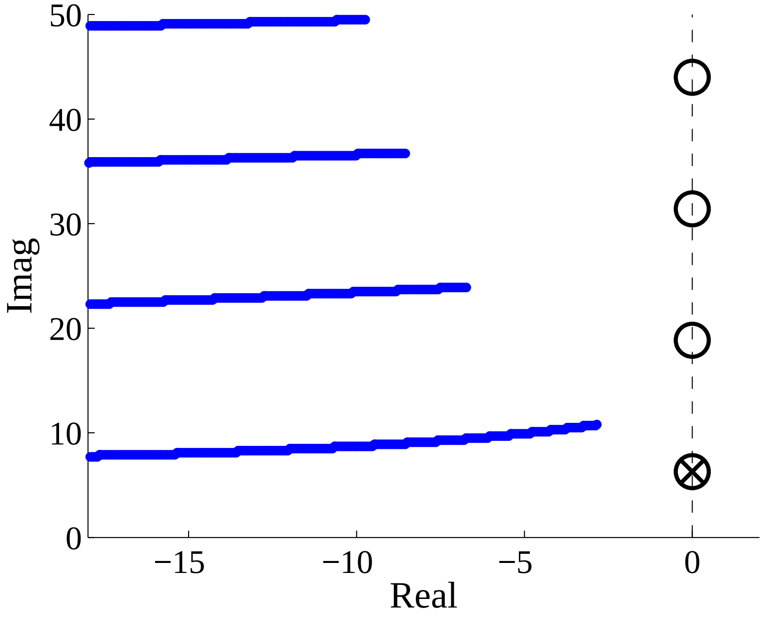


Figure 6.5: Root Locus with exact pole/zero cancelation.

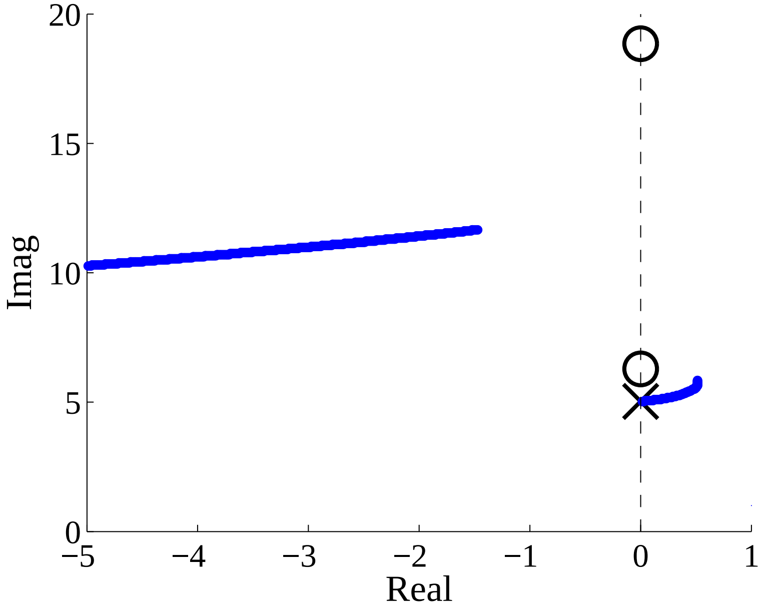


Figure 6.6: Root Locus where $\omega_a < \omega_m$.

If a modeling error occurs, then the pattern depicted in Figure 6.4 emerges. If $\omega_a < \omega_m$, then the root locus is as shown in Figure 6.6. Clearly, the root locus branch extending from the plant poles goes instantly unstable. However, if $\omega_a > \omega_m$, this branch remains stable. This can be seen in the Figure 6.7. This pattern matches the

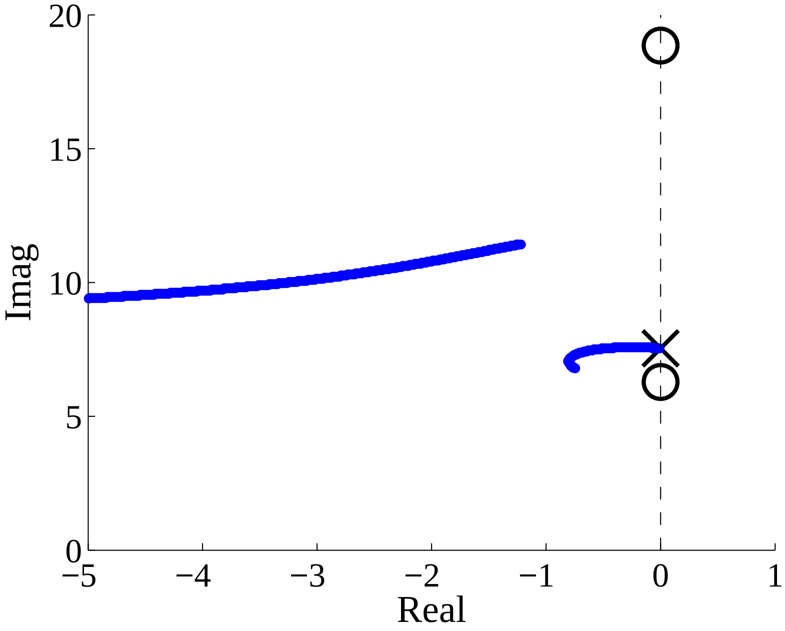


Figure 6.7: Root Locus where $\omega_a > \omega_m$.

prediction made in Figure 6.4. Note that Staehlin and Singh had similar stability results for their CLSS controller which will be discussed in Chapter 8 [91].

6.2.2.2 Bode Analysis of Classical Method

The basic control system (again, C is just a proportional controller K) is now analyzed via the Bode diagram. The Bode diagram of the second-order system used in (6.5) is shown in Figure 6.8. Again, the plant parameters are $\omega_a = 2\pi$ and $\zeta = 0$. Furthermore, the Bode diagram of an input shaper tuned to this plant's frequency and damping ratio can be seen in the Figure 6.3. The Bode diagram of the entire open-loop system depicted in Figure 6.1 (setting $C = K = 1$) is shown in Figure 6.9. Here, the plant peak is exactly canceled by the shaper trough.

Next, the same two cases of modeling error investigated via the root locus plot ($\omega_a < \omega_m$ and $\omega_a > \omega_m$) are analyzed with Bode diagrams. If $\omega_a < \omega_m$, the Bode plot looks like that shown in Figure 6.10. This system has negative phase and gain margins. However, if $\omega_a > \omega_m$, then the frequency plot has positive phase and gain

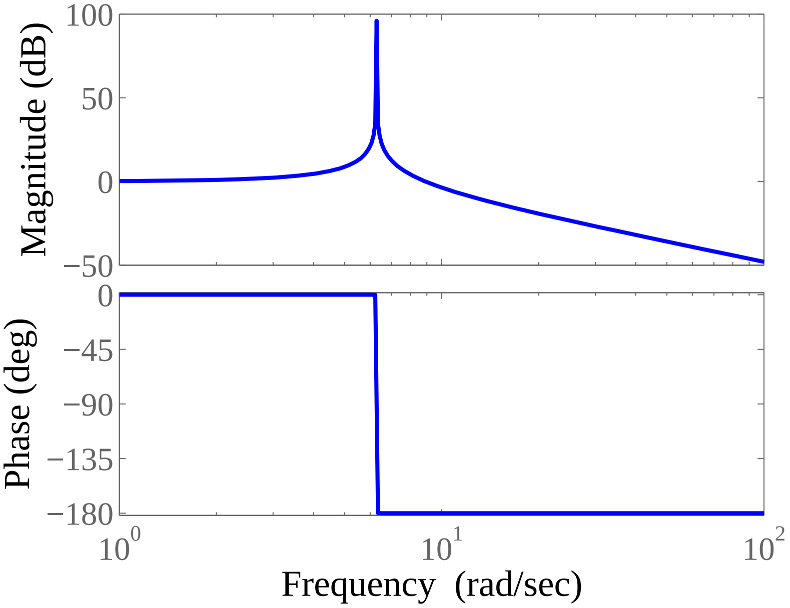


Figure 6.8: Bode Diagram of an Undamped, 2^{nd}-Order Plant.

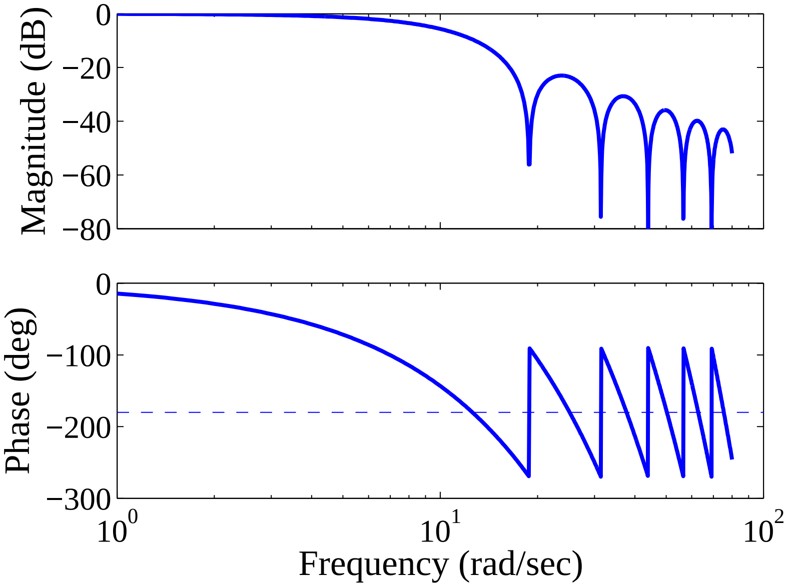


Figure 6.9: Bode Diagram of Complete, Open-Loop System.

margins as shown in Figure 6.11. The same stability trend previously shown via root locus plots is reiterated here using Bode plots.

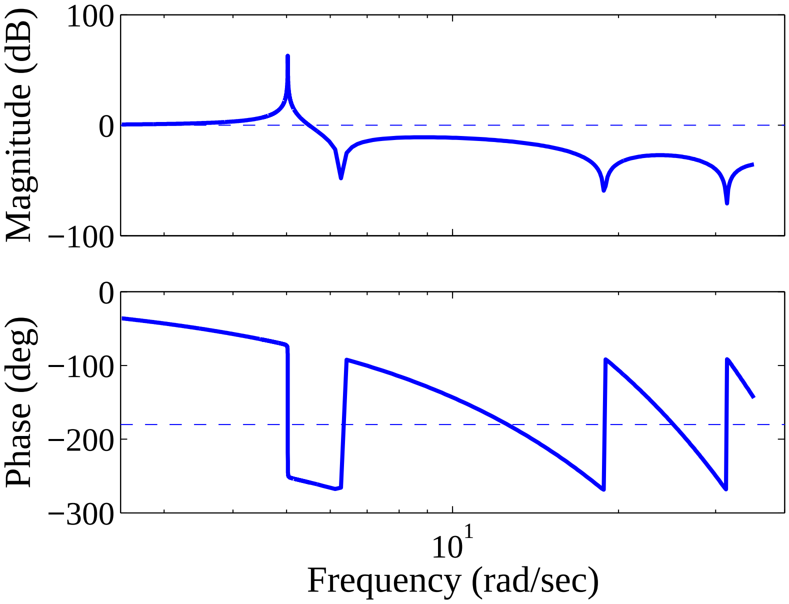


Figure 6.10: Open Loop Bode when $\omega_a < \omega_m$.

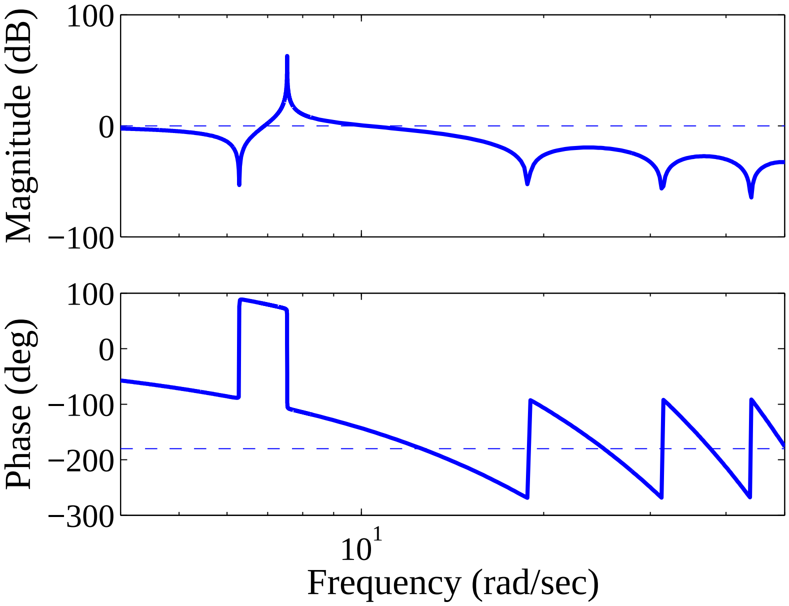


Figure 6.11: Open Loop Bode when $\omega_a > \omega_m$.

6.2.2.3 Parameter Influence on Closed-Loop Stability

The previous sections established the basic form of the root locus and Bode plots for the Classical Method form of CLSS. Also, the previous sections revealed that modeling errors are a primary source of instability for CLSS controllers. Here, the

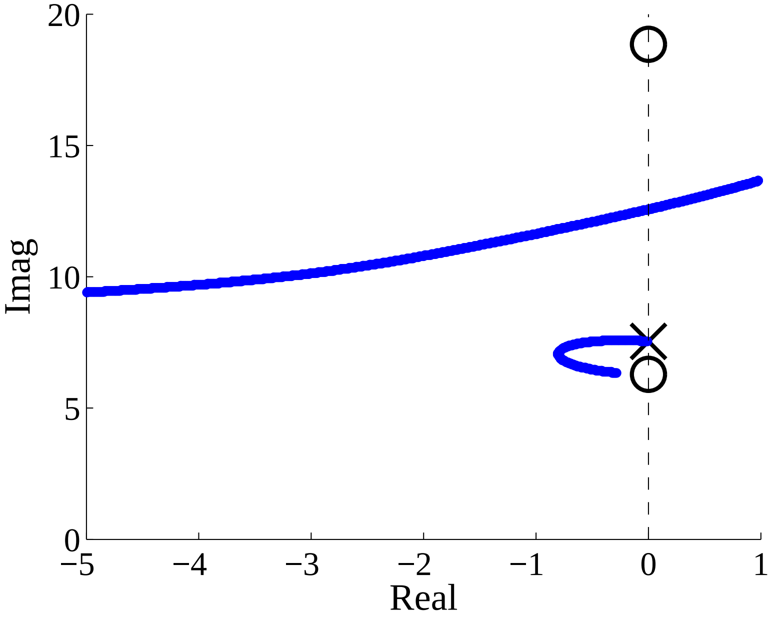


Figure 6.12: Root Locus with high K value.

influence of other parameters (namely K and ζ) that are present in the control system (Figure 6.1) will be analyzed. Also, the addition of a lead compensator to the open-loop system is investigated. The lead compensator is studied here because it is known to increase a system's stability margins. Because small modeling errors can result in unstable, CLSS controllers, it was important to show that a control block (C) could be added to the block diagram shown in Figure 6.1 so as to increase stability margins even when modeling errors occur.

Influence of Gain, K

By increasing the gain, K, the system shown in Figure 6.1 will eventually be driven unstable. Figure 6.12 shows how an initially stable system can be driven unstable by making K too high. This figure indicates that the dynamics arising from the inclusion of input shaping filters inside of feedback loops can, themselves, be the cause of instability.

This same phenomenon is reiterated via a Bode plot in Figure 6.13. In this figure,

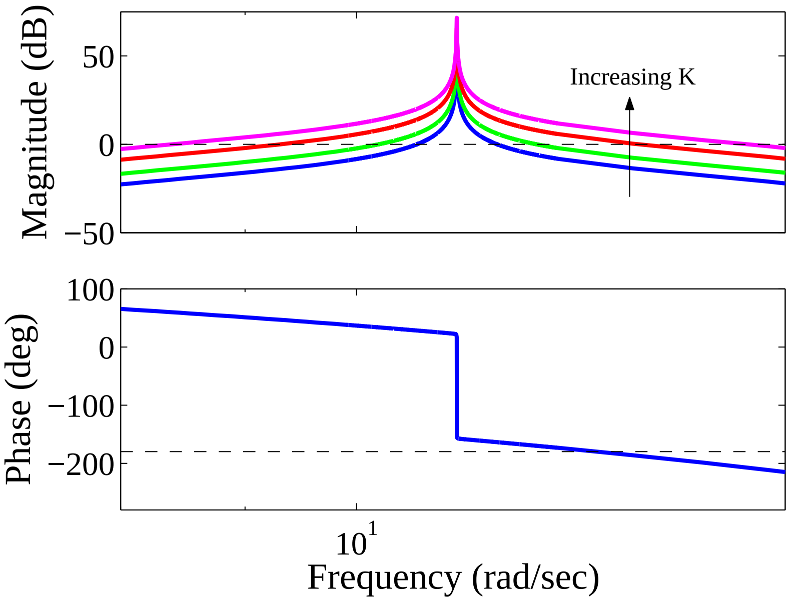


Figure 6.13: Influence of K on Bode Diagram of a CLSS Controller.

the rising magnitude plot is caused by the increasing K value. Eventually, this causes both stability margins to change from positive to negative.

Influence of damping ratio, ζ

However, when the second-order plant has damping, this increases the regions of stability. Figure 6.14 shows this effect on a root locus plot. By increasing ζ, the root locus branches are shifted to the left. This effect enables higher proportional gain (K) values to result in stable closed-loop poles. Figure 6.15 shows the effect of increasing ζ on a Bode plot. By flattening both the magnitude and phase plots, the phase margin and gain margin are increased.

Influence of Lead Compensator

Because of the stability issues associated with closed-loop signal shaping, it was desirable to study the effect of stability enhancing controllers. One of the most effective and practical controllers used to increase stability margins is the lead compensator.

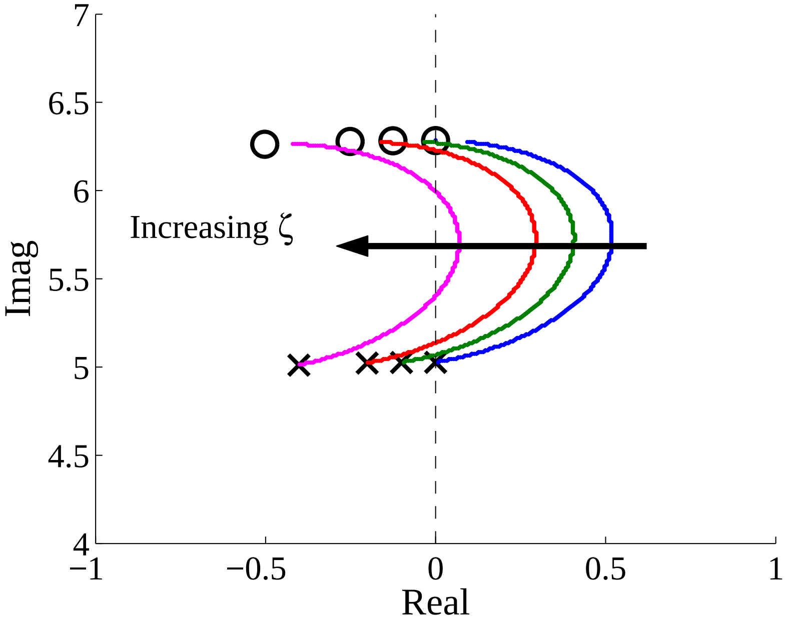


Figure 6.14: Influence of ζ as shown on root locus plot.

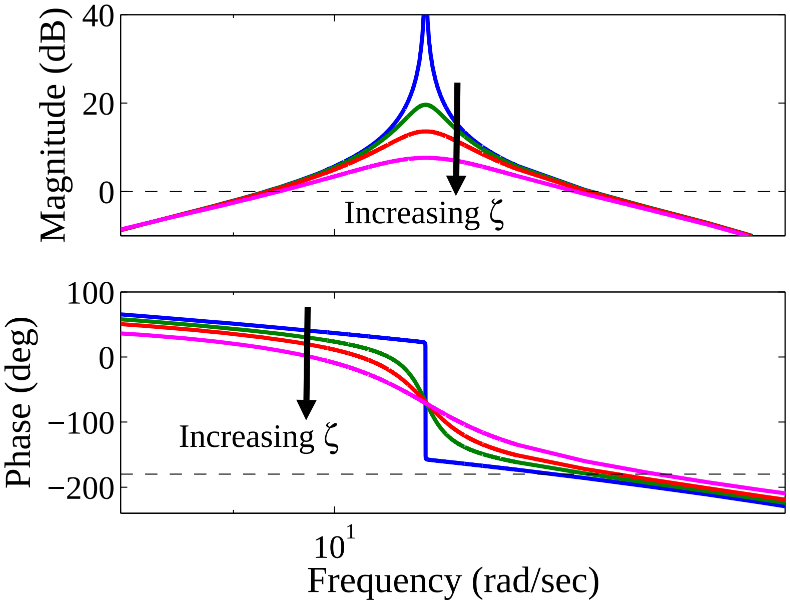


Figure 6.15: Influence of ζ as shown on Bode plot.

The lead compensator is implemented here by setting the controller, C, to $C = K\frac{s+z}{s+p}$. Figure 6.16 shows how the root locus branches are pulled to the left and Figure 6.17 shows how phase and gain margins are increased.

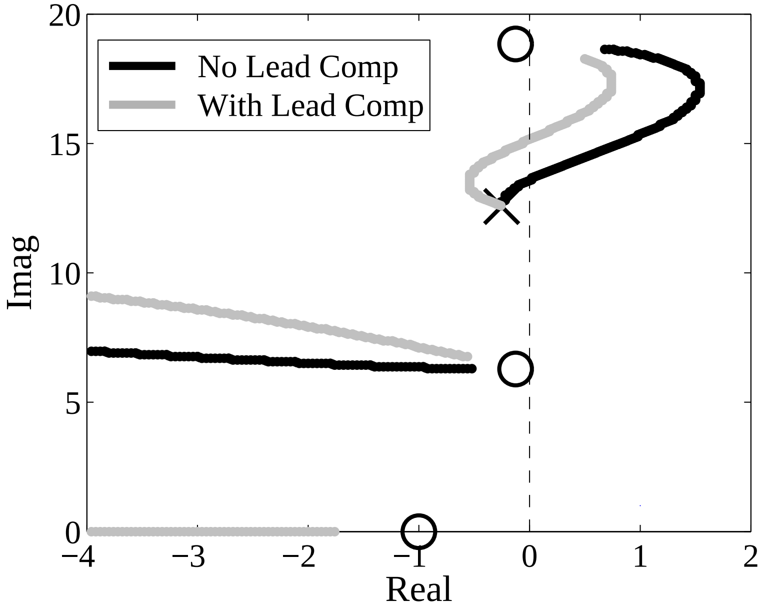


Figure 6.16: Effect of Lead Compensator on Root Locus.

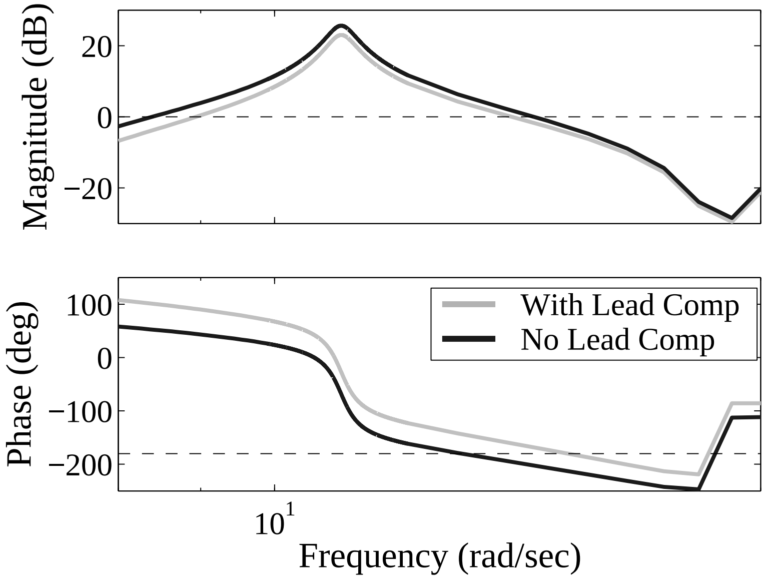


Figure 6.17: Effect of Lead Compensator on Bode Diagram.

While there are many other types of controllers (PID, lag compensator, etc.) that could have been used as the C block in Figure 6.1, the main goal of this section was to show that reasonable stability margins can be achieved in CLSS systems. The lead compensator sufficiently proved this point. However, because the basics of the root locus and Bode diagrams for CLSS controllers have now been established and reviewed, the effect of any control block (C) on the Classical Method of CLSS controllers can be studied.

6.3 Closed-Loop Signal Shaping: Digital Root Loci

Another way to view the closed-loop poles of a feedback control system is with the digital root locus. The main advantage for using the digital root locus in CLSS design is that an input shaper in the digital domain is a finite-order polynomial in z. This eliminates the need for numerical root locus drawing techniques and eliminates the presence of an infinite number of open-loop zeros and closed-loop poles. This section will show several CLSS digital root loci and highlight some of the same trends found in the continuous domain root loci that are fundamental to CLSS design.

6.3.1 Basic Effect of Input Shapers on Digital Root Loci

This section will analyze the digital root loci of the block diagram depicted in Figure 6.1 when $C = K$, $Plant = 1$ and "I" is a ZV shaper. This analysis will yield a basic understanding of the effect that an input shaper has when included within the feedback loop, similar to the results shown in Figures 6.2 and 6.3.

The control scheme whose digital root locus is shown in Figure 6.18 has a sampling time (T) equal to one half the ZV shaper's duration ($T = \frac{\pi}{2\omega_d}$). As shown in Figure 1.4, the duration of an input shaper (Δ) is equal to the time of the last impulse. Figure 6.19 shows the digital root locus of the same system when $T = \frac{\pi}{3\omega_d}$, while Figure 6.20 shows the digital root locus of the same system when $T = \frac{\pi}{8\omega_d}$. As T becomes smaller, the digital domain input shaper has an increasing number of open-loop zeros and poles. As can be seen from these figures, as T approaches zero, the input shaper's zeros and poles approach an infinite number. This is consistent with earlier results because as a digital control system's T approaches zero, the system approaches a continuous domain system.

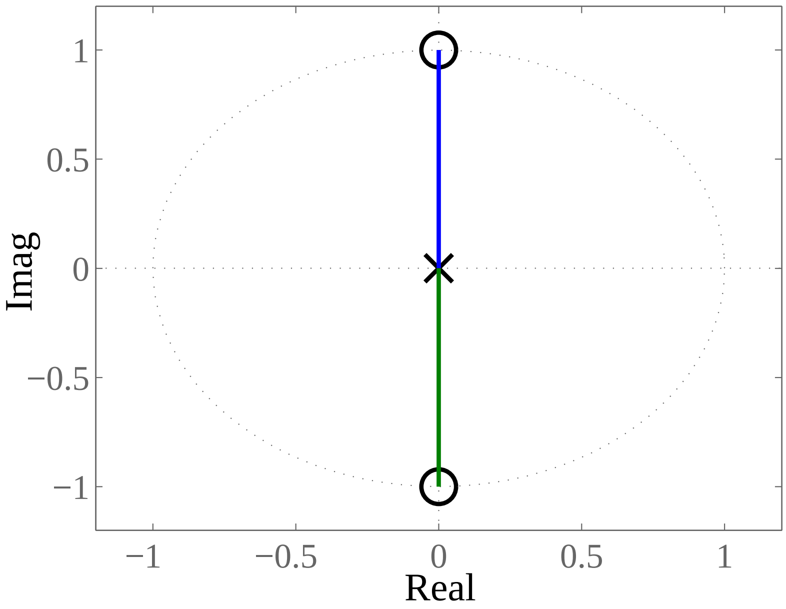


Figure 6.18: ZV Shaper for $\zeta = 0$, Shaper Duration = 2*T.

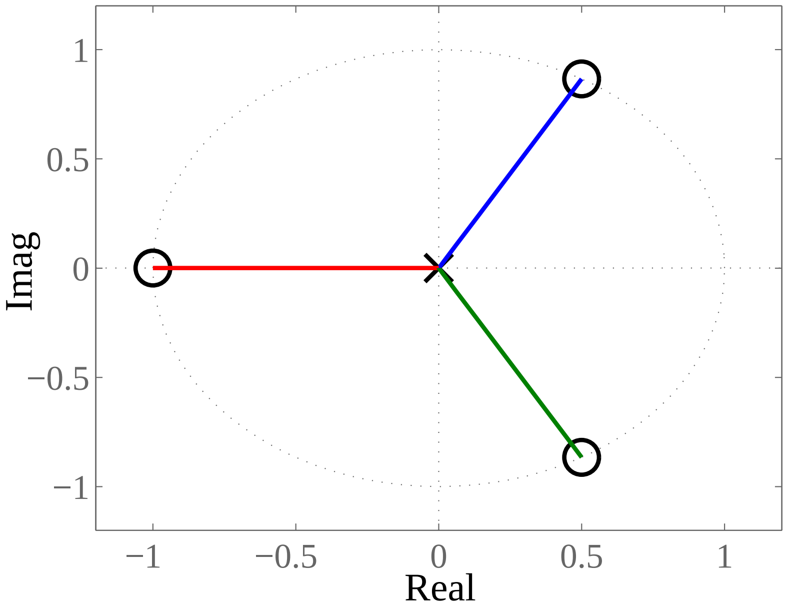


Figure 6.19: ZV Shaper for $\zeta = 0$, Shaper Duration = 3*T.

Also notice that the open-loop zeros of this input shaper always lie on the unit circle. This is consistent with the continuous domain root loci, where the open-loop zeros of an input shaper designed for undamped dynamics lie on the imaginary axis.

Figure 6.21 shows the effect of non-zero damping. In this case, the open-loop zeros

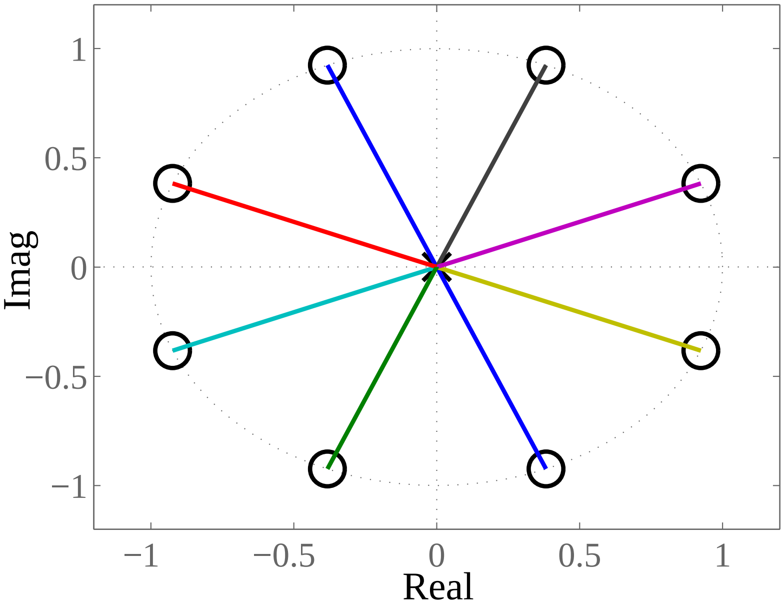


Figure 6.20: ZV Shaper for $\zeta = 0$, Shaper Duration = 8*T.

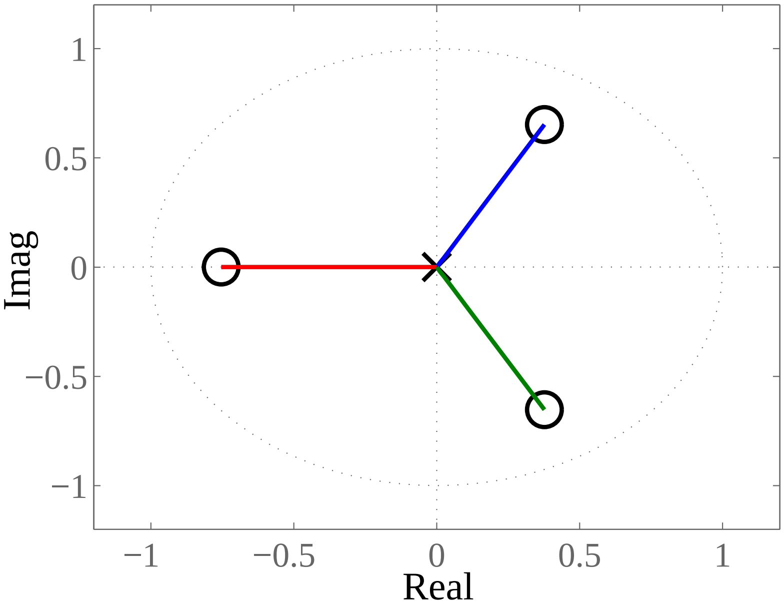


Figure 6.21: ZV Shaper for $0 < \zeta < 1$, Shaper Duration = 3*T.

of the input shaper lie inside the unit circle. This is consistent with the fact that the open-loop zeros of continuous domain input shapers, designed to cancel damped poles, lie to the left of the imaginary axis. Figure 6.22 shows the digital root locus when the shaper duration is not an exact multiple of the sampling time. In this case, two

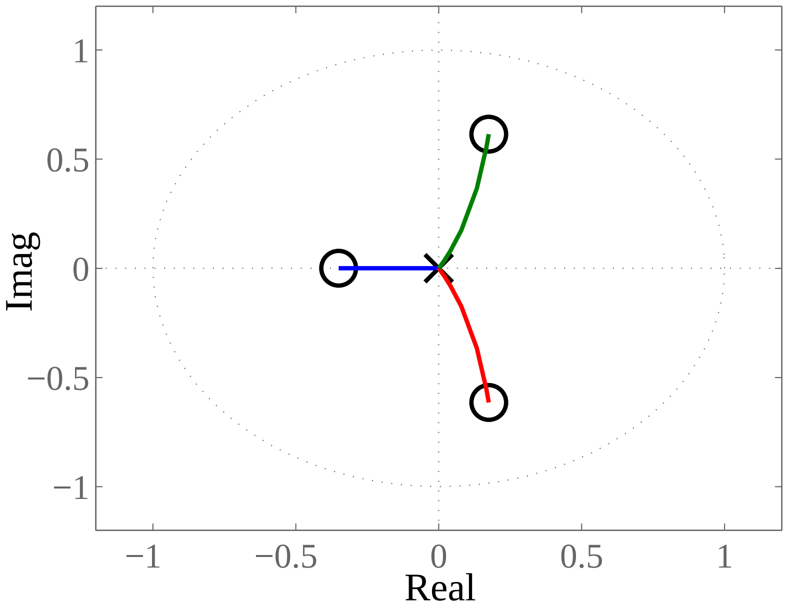


Figure 6.22: ZV Shaper for $0 < \zeta < 1$, Shaper Duration $\neq$ 3*T.

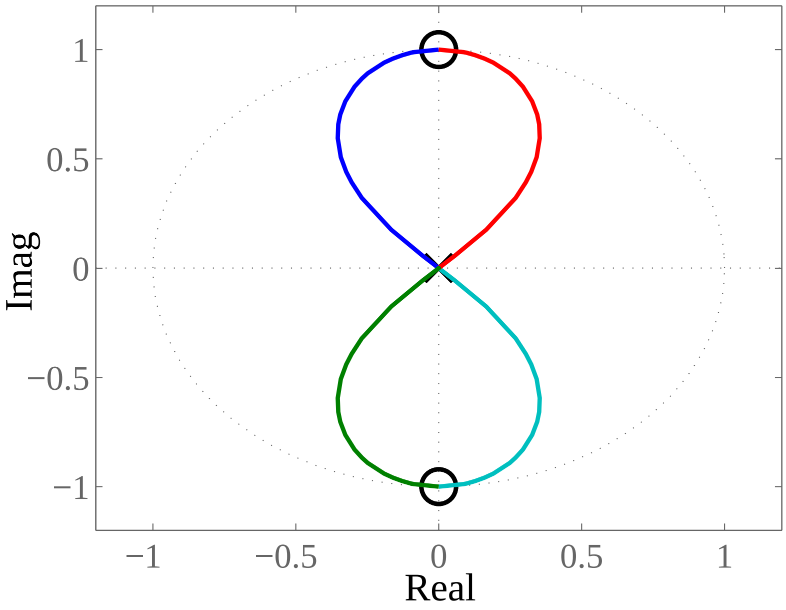


Figure 6.23: ZVD Shaper for $\zeta = 0$, Shaper Duration = 4*T.

of the open-loop zeros lie at different locations (as compared to Figure 6.21, where the shaper duration was an exact multiple of the sampling time) and the normally straight lines from pole to zero are slightly curved.

Figure 6.23 shows the digital root locus when a ZVD shaper is utilized within the

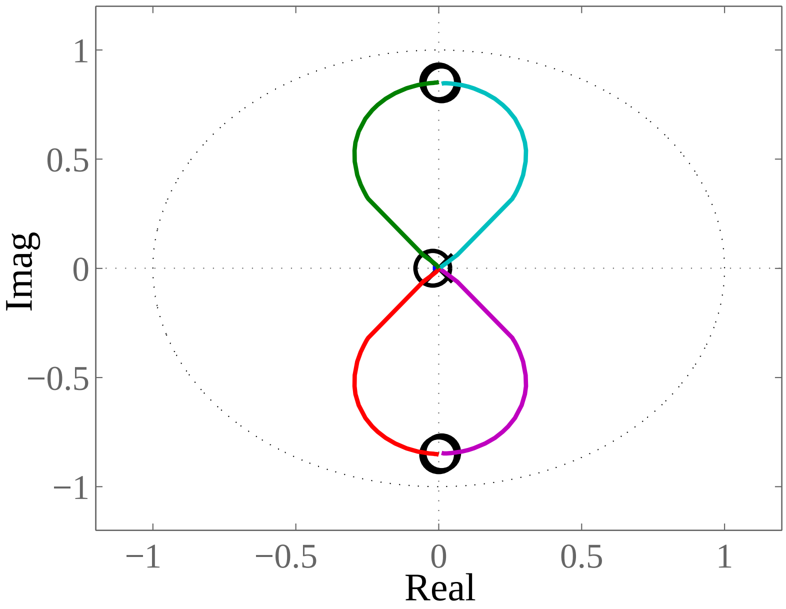


Figure 6.24: ZVD Shaper for $0 < \zeta < 1$, Shaper Duration $\neq$ 4*T.

feedback loop. This input shaper has two open-loop zeros/poles at each place where a ZV shaper would have only one. Again, the effect of damping is similar to that on a ZV shaper, as shown by Figure 6.24.

6.3.2 CLSS Digital Root Loci with Non-Unity Plants

The combination of an input shaping controller within the feedback loop and a non-unity plant will now be investigated. Here, the plant is assumed to be a second-order oscillator. A ZV shaper exactly tuned to cancel the undamped poles of a second-order system will yield the digital root locus shown in Figure 6.25. Just as in the continuous domain root locus, even when perfect pole/zero cancelation occurs, the system will go unstable for some gain value. The point of instability is shown as a small “x” or “+” on the root locus.

This instability problem is exasperated by modeling errors and imperfect pole/zero cancelation. When the modeled plant frequency (used to design the input shaper) is less than the actual plant frequency, the root locus is as shown in Figure 6.26. This system goes unstable quicker than with perfect modeling, but still has a reasonable

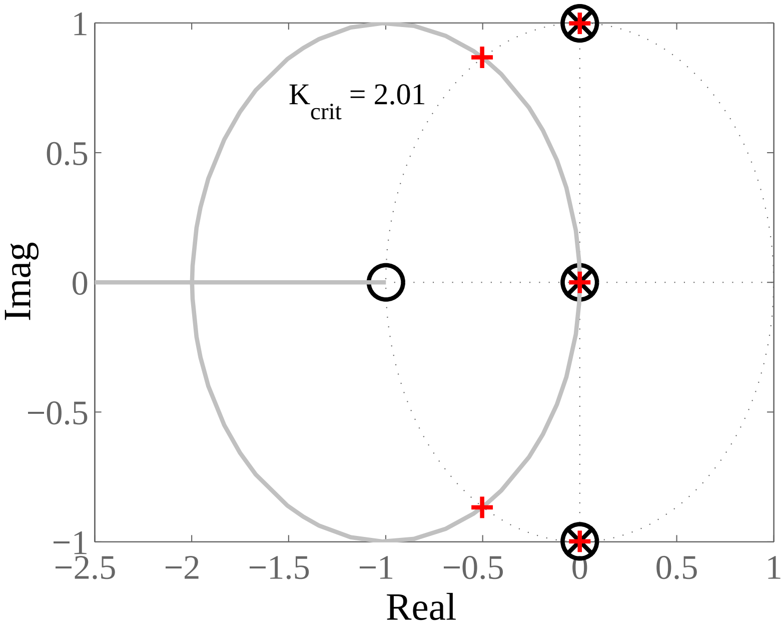


Figure 6.25: Digital Root Locus, ω_m=ω_a.

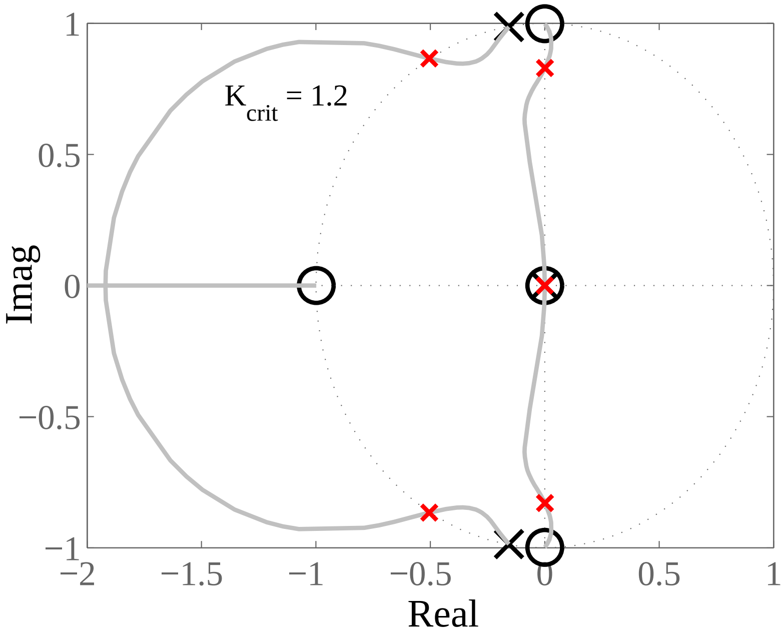


Figure 6.26: Digital Root Locus, $\omega_m < \omega_a$.

range of system gain in which stability holds. However, when the modeled frequency is higher than the actual frequency, the root locus shown in Figure 6.27 results. Here, instability is virtually immediate, with relatively small system gains driving

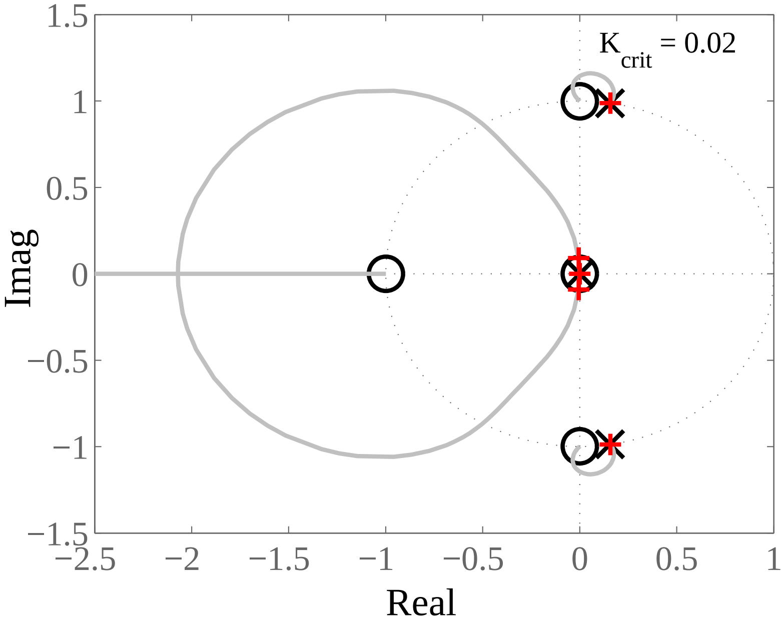


Figure 6.27: Digital Root Locus, $\omega_m > \omega_a$.

the system unstable. Note that this relationship between modeled and actual natural frequency was predicted by the continuous time root loci as well.

One way to enhance stability margins is to decrease the system's sampling time. Figure 6.28 shows a digital root locus of the same system depicted in Figure 6.25 with the sampling time cut in half. Here, the system gain which results in instability is slightly higher than in the system shown in Figure 6.25.

Finally, Figure 6.29 shows the combined affect of system damping and system sampling time on the system gain resulting in instability (K_{crit}). Here, the open-loop system consisted of a ZV shaper that was exactly tuned to cancel the poles of an undamped, second-order plant. As can be seen from Figure 6.29, increasing the plant's damping ratio increases stability margins. Note that this effect was predicted in the continuous domain root loci. Also, decreasing the sampling period generally increases the stability margins.

Note from Figure 6.29 that very low damping ($\zeta = 0.01$ and $\zeta = 0.001$) combined with relatively high sampling periods tended to create jumps in the K_{crit} curves. The

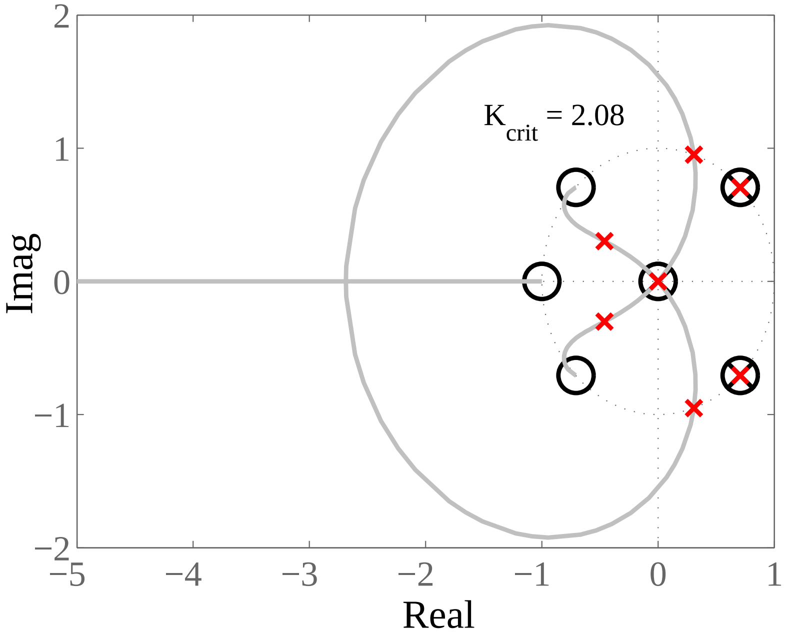


Figure 6.28: Digital Root Locus, $\omega_m = \omega_a$, Smaller T.

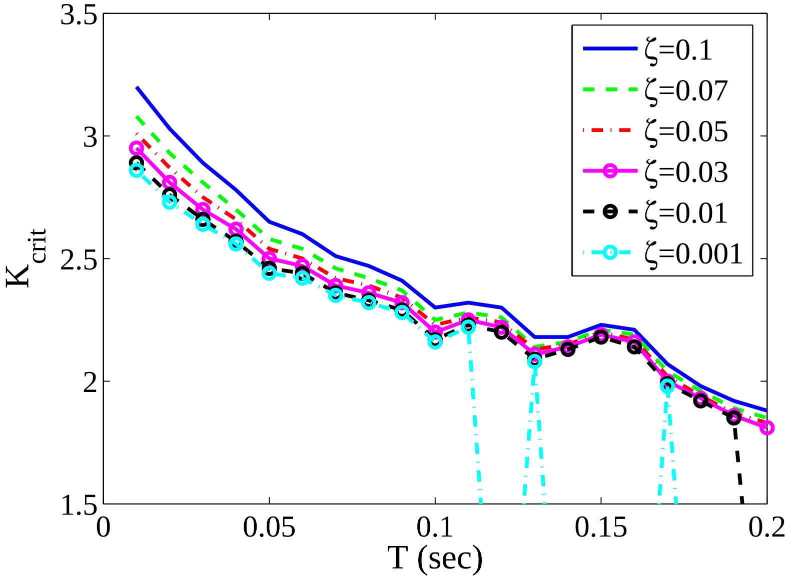


Figure 6.29: Effect of T and ζ on K_{crit}: $\omega_m = \omega_a = 2\pi$.

jumps toward $K_{crit} \approx 0$ are caused by the input shaper zero not fully canceling the plant pole. For example, the digital root locus of the system shown in Figure 6.1 with $\zeta = 0.001$ and $T = 0.15sec$ is shown in Figure 6.30. Figure 6.31 shows a zoomed in view of Figure 6.30 using the boundaries approximated by the "Zoom" box labeled on

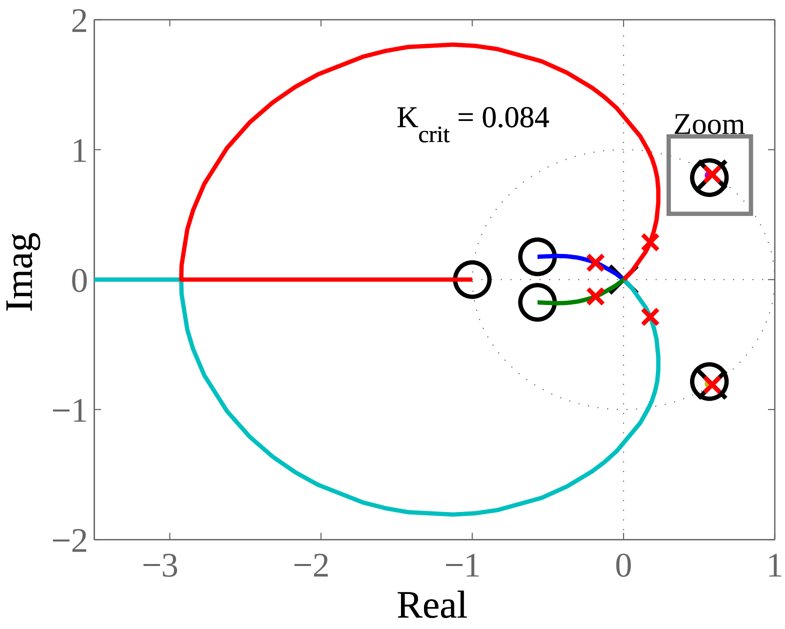


Figure 6.30: Digital Root Locus: $\zeta = 0.001$ and $T = 0.15sec$.

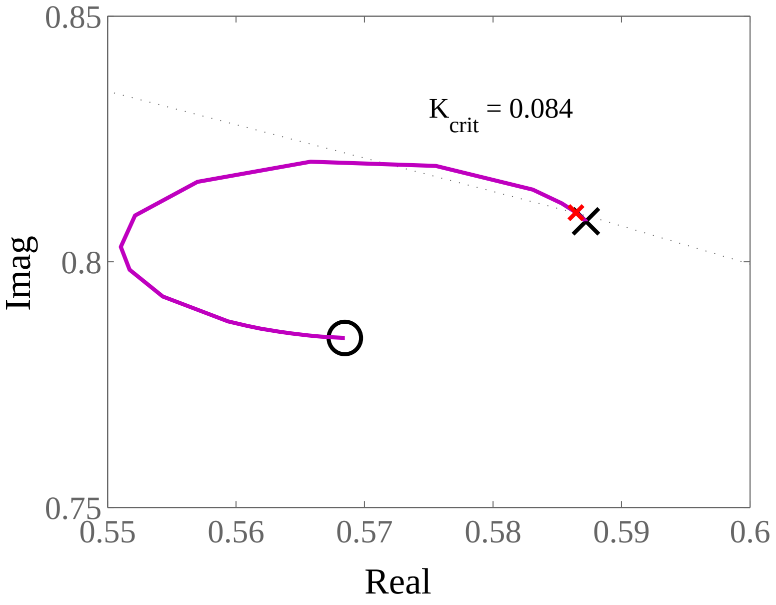


Figure 6.31: Zoomed in Digital Root Locus: $\zeta = 0.001$ and $T = 0.15sec$.

Figure 6.30. Because the sampling period is large relative to the shaper duration (the shaper duration is $0.5sec$ in all of the digital root locus figures shown in this section), the pole/zero cancelation is not completely accomplished. This causes a short root

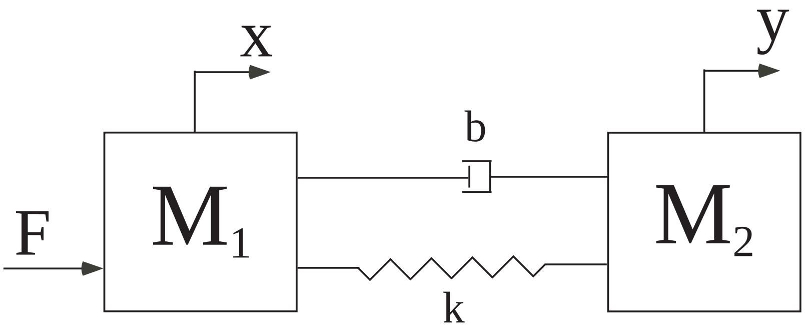


Figure 6.32: Mass-Spring-Mass System.

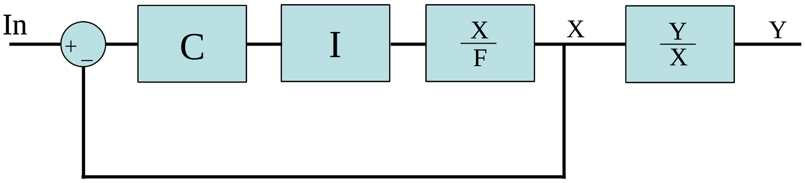


Figure 6.33: Block Diagram of Collocated Control System.

locus branch to exist. And, because the plant damping is low, this branch exists close to (and in this case crosses) the unit circle. Either decreasing T or increasing ζ will generally solve this problem.

6.4 Mass-Spring-Mass System

In order to investigate more complicated systems, a mass-spring-mass system was studied. In particular, this mass-spring-mass system could represent a satellite, or spacecraft, which has a rigid body connected to a lightweight, flexible appendage. A schematic of the mass-spring-mass system is shown in Figure 6.32. A block diagram of the collocated control system is shown in Figure 6.33. Analyzing the system shown in Figure 6.32, $\frac{X}{F}$ and $\frac{Y}{X}$ can be shown to be:

$$\frac{X}{F} = \frac{M_2 s^2 + bs + k}{s^2 \left(M_1 M_2 s^2 + b[M_1 + M_2]s + K[M_1 + M_2]\right)} \tag{6.6}$$

$$\frac{Y}{X} = \frac{bs + k}{M_2 s^2 + bs + k} \tag{6.7}$$

These equations can be rewritten as:

$$\frac{X}{F} = \frac{1}{M_1} \frac{s^2 + 2\zeta_2\omega_2 s + \omega_2^2}{s^2(s^2 + 2\zeta_1\omega_1 s + \omega_1^2)} \tag{6.8}$$

$$\frac{Y}{X} = \frac{2\zeta_2\omega_2 s + \omega_2^2}{s^2 + 2\zeta_2\omega_2 s + \omega_2^2} \tag{6.9}$$

where:

$$\omega_1 = \sqrt{K\frac{M_1 + M_2}{M_1 M_2}} \tag{6.10}$$

$$\omega_2 = \sqrt{\frac{K}{M_2}} \tag{6.11}$$

$$2\zeta_1\omega_1 = b\frac{M_1 + M_2}{M_1 M_2} \tag{6.12}$$

$$2\zeta_2\omega_2 = \frac{b}{M_2} \tag{6.13}$$

For this analysis, the following values were chosen: $M_1 = 10$, $M_2 = 1$, $k = 20$ and $b = 0.2$. These values lead to: $\omega_1 = 4.69\frac{rad}{s}$, $\omega_2 = 4.47\frac{rad}{s}$, $\zeta_1 = 0.024\frac{rad}{s}$, $\zeta_2 = 0.02\frac{rad}{s}$.

6.4.1 Bode Diagram Analysis

In order to study the stability of the system shown in Figure 6.33, the open-loop Bode diagram and the root locus are analyzed. If it is assumed that the controller, C, is just a proportional controller, then the open-loop Bode diagram of $KI\frac{X}{F}$ (where $K = 1$ and I is a ZV shaper) would look like that shown in Figure 6.34. Unfortunately, it can be easily seen that the phase margin is negative.

One possible solution would be to greatly increase K so that the phase is above $-180°$ when the gain crosses the $0dB$ line. However, Figure 6.35 (which shows the same open-loop Bode, but with $K = 2500$) shows that the system is still unstable.

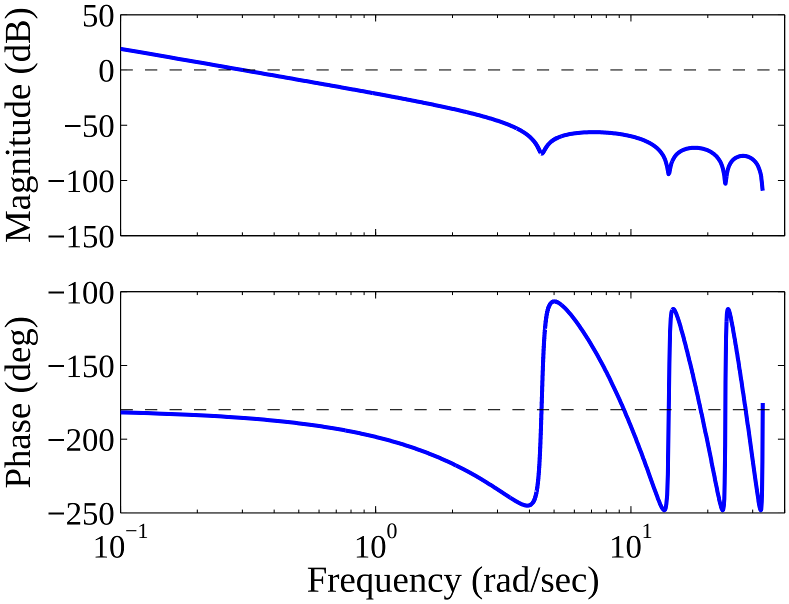


Figure 6.34: Open-Loop Bode Diagram.

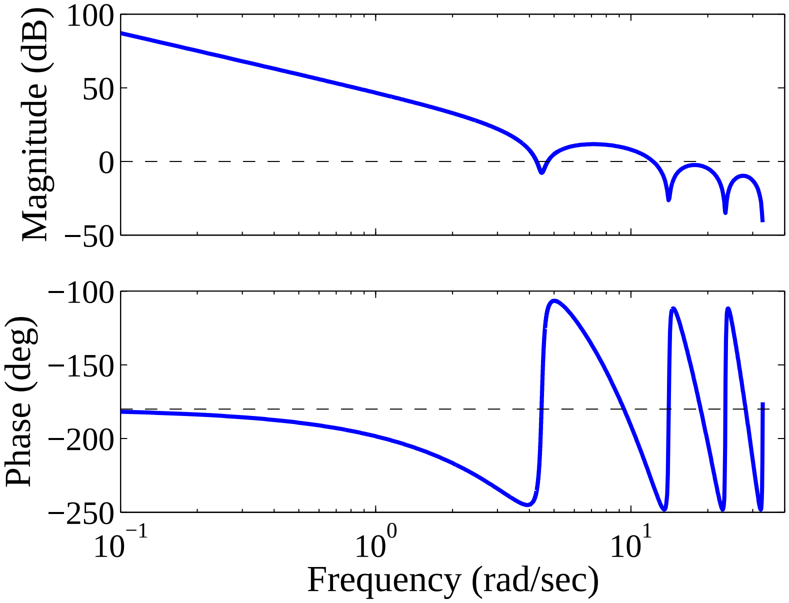


Figure 6.35: Open-Loop Bode Diagram with $K = 2500$.

This Bode plot has 3 gain crossover frequencies, and the first of these still has a negative phase margin. Even if K is increased further (so that the first dip in the magnitude curve did not create any crossovers), the next crossover would also have a negative phase margin.

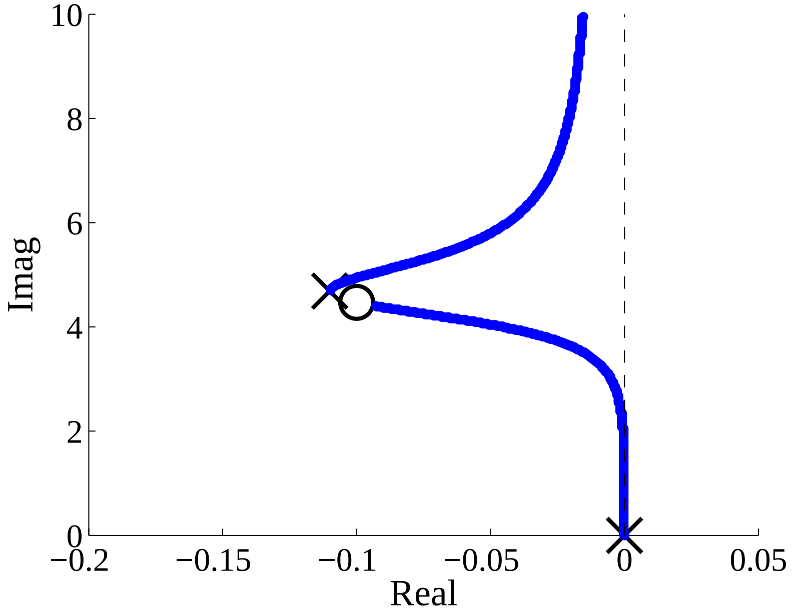


Figure 6.36: Root Locus of Control System without Input Shaper.

6.4.2 Root Locus Analysis

The root locus of the closed-loop system shown in Figure 6.33, **without** the input shaper, is shown in Figure 6.36. This system is stable for any K value. However, if the input shaper is added to the closed-loop system, then the root locus becomes as shown in Figure 6.37. This system starts out (with small K values) as unstable. And, even when large K values bring the closed-loop poles from the first root locus branch back to stable values, other branches have already gone unstable. This confirms the original findings in the Bode diagram analysis: the CLSS control of a mass-spring-mass system is always unstable. That is, the system is always unstable when the controller, C, is just proportional control.

One solution is to use a lead compensator, in addition to proportional control (i.e. $C = K\frac{s+z}{s+p}$). If the lead compensator's pole is chosen to be at -20 while the zero is chosen to be located at -1, then the system root locus will look like the one shown in Figure 6.38. Clearly, this system is much more stable. Even with the gain, K, taken as high as 3000, the system remains stable. In conclusion, by adding a lead compensator,

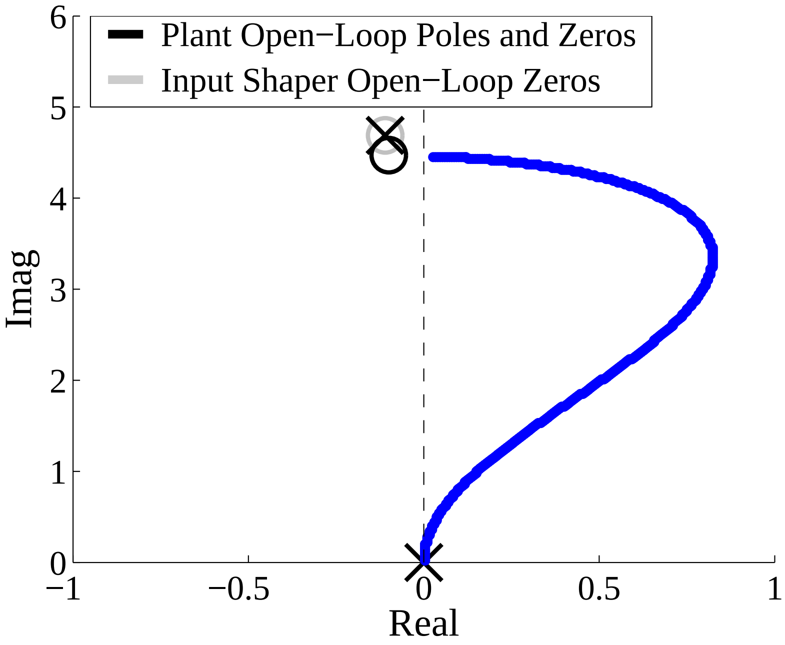


Figure 6.37: Root Locus of Complete Control System.

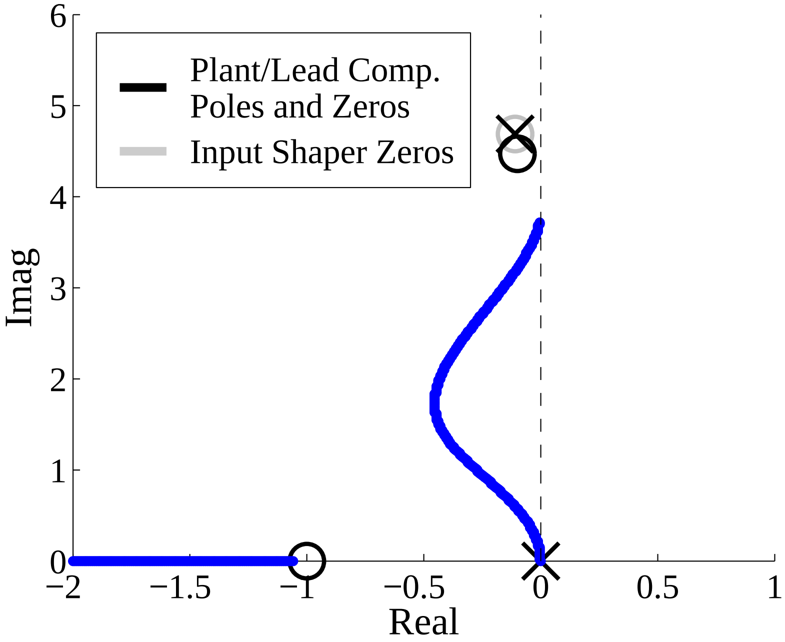


Figure 6.38: Root Locus of Complete Control System with a Lead Compensator.

the closed-loop signal shaping of a fourth-order system is relatively stable. This is further verified by the Bode diagram of the system with a lead compensator, shown in Figure 6.39. Here, the gain margin is large, around $70dB$.

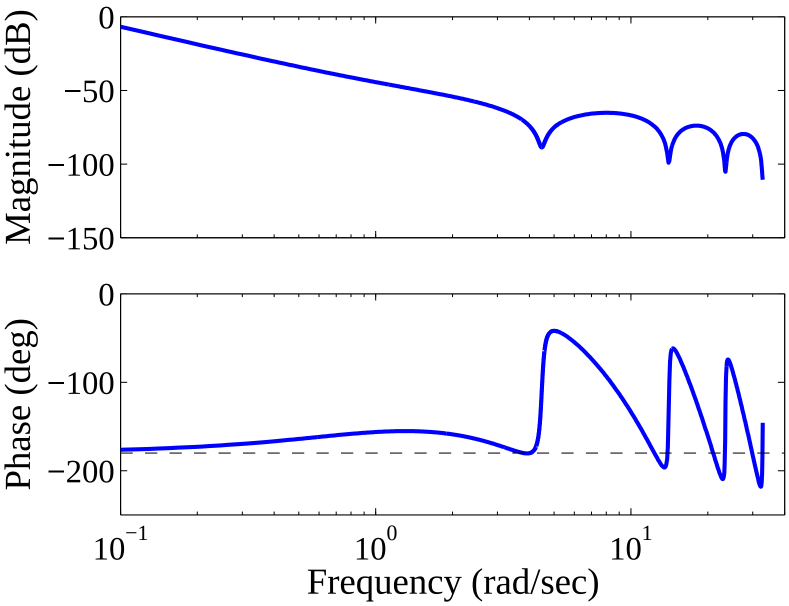


Figure 6.39: Open-Loop Bode Diagram with Lead Compensator.

6.4.3 Modeling Errors

To verify the general stability of closed-loop signal shaping of a fourth-order system with a lead compensator, the affect of modeling errors will be investigated via the root locus. Remember that the plant transfer function for this mass-spring-mass system is:

$$\frac{X}{F} = \frac{1}{M_1}\frac{s^2 + 2\zeta_2\omega_2 s + \omega_2^2}{s^2(s^2 + 2\zeta_1\omega_1 s + \omega_1^2)} \tag{6.14}$$

If the complex poles of $\frac{X}{F}$ are perfectly modeled, then the ZV shaper could theoretically be tuned such that full pole/zero cancelation occurs. This was shown in Figures 6.37 and 6.38. However, this exact modeling is never possible. If the shaper frequency is smaller than the plant frequency, then the root locus shown in Figure 6.40 results. As expected from the earlier study of second-order systems, this scenario results in a root locus branch from the plant pole to the shaper zero that bends to the left. This scenario usually does not greatly decrease stability. However, if the shaper frequency is larger than the plant frequency, then the new root locus branch bends to the right, as shown in Figure 6.41. With second-order systems, this scenario usually causes

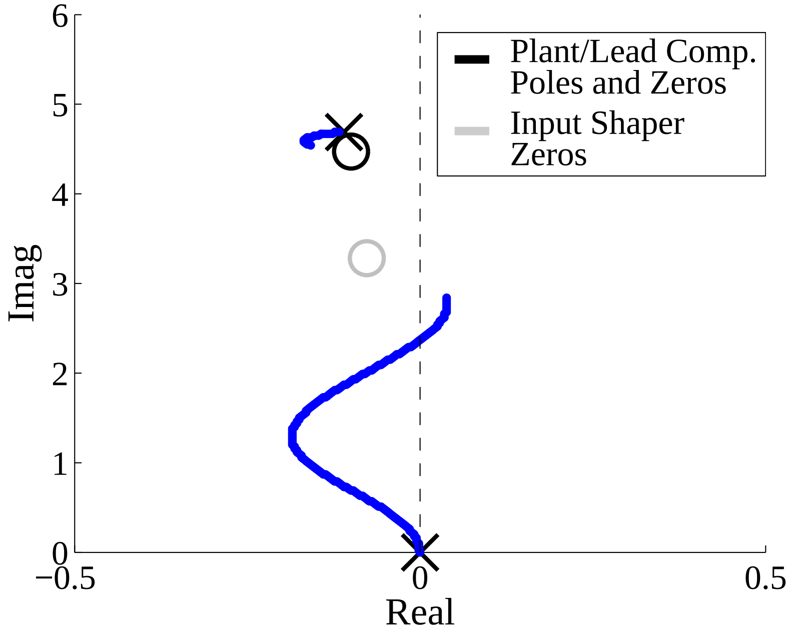


Figure 6.40: Root Locus of Complete Control System with $\omega_m < \omega_1$.

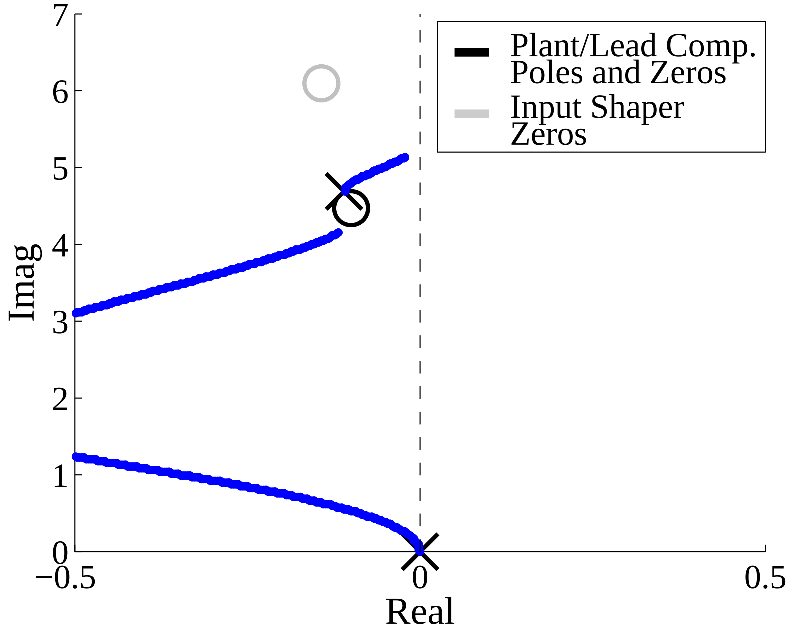


Figure 6.41: Root Locus of Complete Control System with $\omega_m > \omega_1$.

significant stability problems. However, in this fourth-order system, the plant pole is now located between two zeros - one from the input shaper and one from the plant's numerator dynamics. Because it lies between two zeros, the plant pole does not adversely affect stability as much as in the second-order case. According to Figure 6.41,

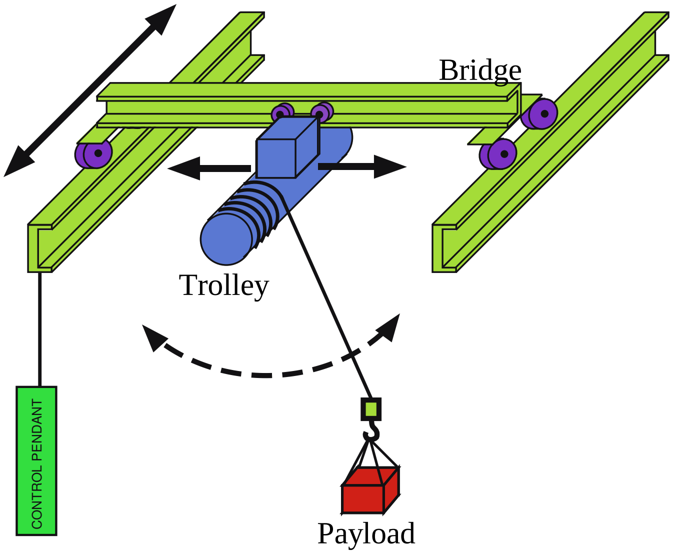


Figure 6.42: Crane Used for Stability Experiments.

even with a K value of 3000, the root locus branch extending from the plant pole remains stable.

6.5 Experimental Verification

In order to verify the stability findings regarding CLSS controllers, experiments were conducted on an industrial crane located at the Georgia Institute of Technology. The crane is a 10-ton bridge crane, sketched in Figure 6.42. This bridge crane has at least 4 notable nonlinearities: a velocity limit, an acceleration limit, a built-in velocity smoothing algorithm that prevents sudden sign changes in velocity, and a velocity dead zone. Obviously, the linear based root locus analysis tool presented in this thesis cannot account for the nonlinear behavior of this real world system. However, it will be shown here that a good linear model is sufficient to explain the dynamic behavior and approximately predict the gain value that induces instability.

The motion of the crane trolley is controlled by a CLSS controller of the form shown in Figure 6.1. Here,

$$C = K_o(K_p + K_d s) \qquad I = A_1 + A_2 e^{-st_2} \tag{6.15}$$

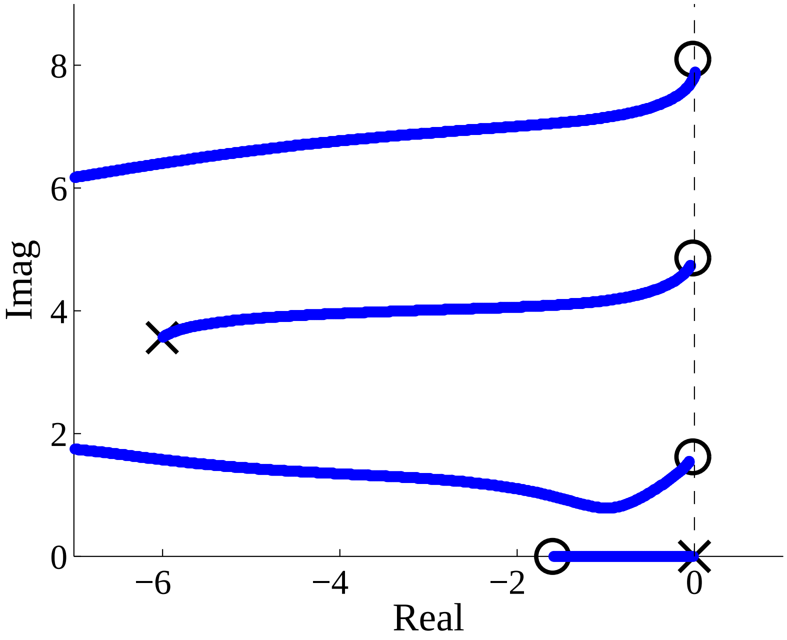


Figure 6.43: Root Locus of Crane Control System.

$$G = \frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}. \tag{6.16}$$

This closed-loop system consists of a PD feedback controller, a two impulse (ZV) shaper, and a third-order plant (second-order oscillator plus an integrator). For these experiments, all values within the closed-loop system were held constant, except K_o. K_o was varied to show its effect on stability. The root locus of this closed-loop system is shown in Figure 6.43. As can be seen from the zoomed-in root locus plot shown in Figure 6.44, this closed-loop system will go unstable for some finite value of gain K_o. In fact, the root locus predicts that a K_o value of 10.5 will create a marginally stable system.

In order to verify these predictions, the crane trolley was commanded to move 1 meter under the closed-loop signal shaping controller described in this section. As an initial test, the experiments were conducted without an input shaper in the loop. That is, I was set to 1 and C and G were left as shown in (6.15) and (6.16). This was done to ensure that the system would not go unstable purely due to its nonlinearities

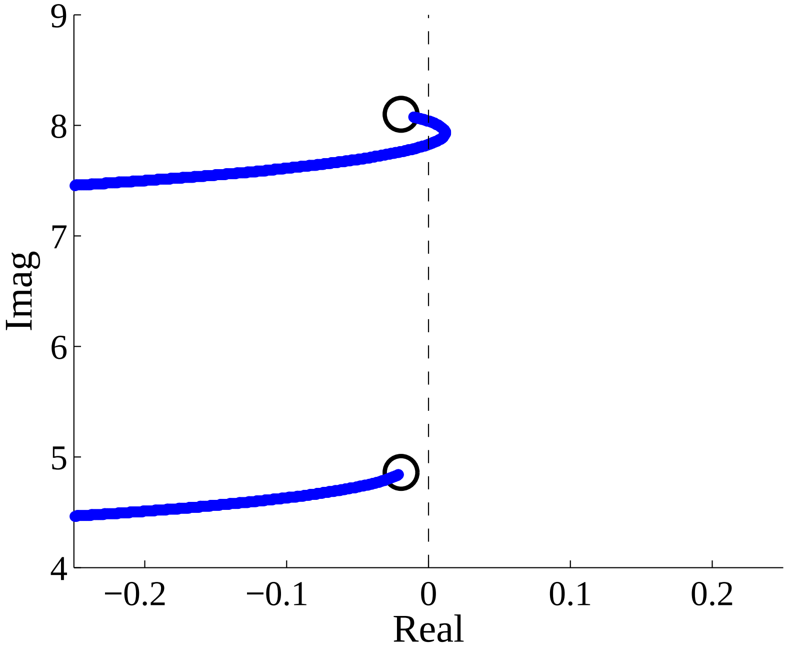


Figure 6.44: Closed-Loop System Goes Unstable.

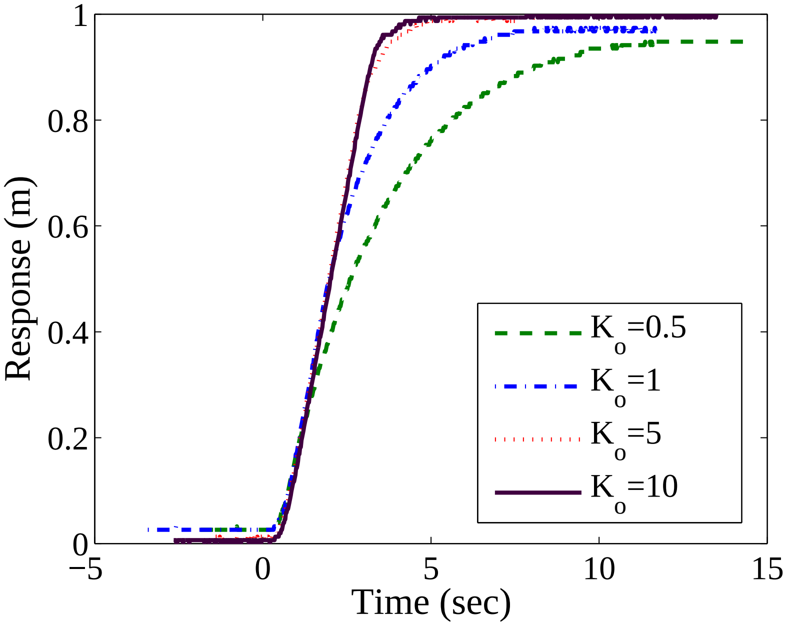


Figure 6.45: Crane Response without CLSS.

and PD controller. As seen in the experimental results shown in Figure 6.45, no value of K_o resulted in instability. However, when the input shaper was included within the feedback loop, increasing K_o from 0.5 to 10 causes a limit cycle, as shown in Figure

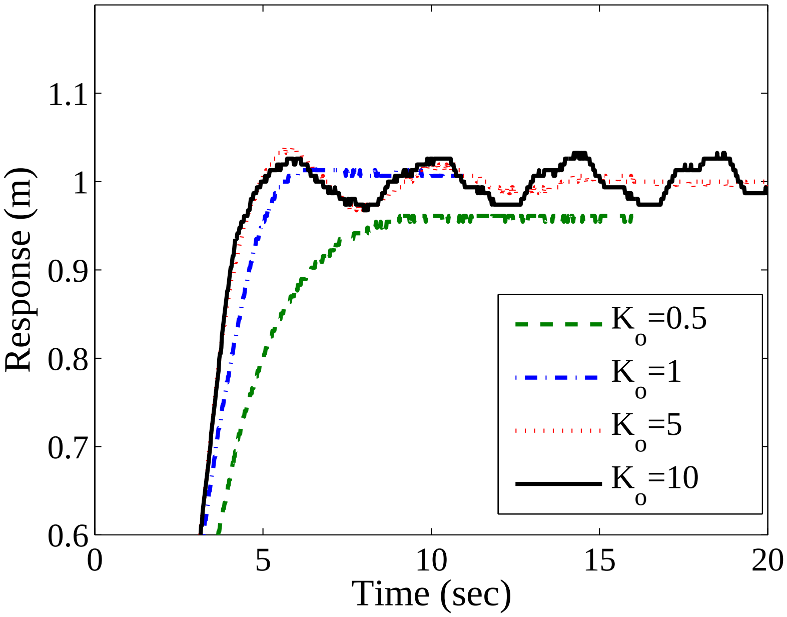


Figure 6.46: Crane Response with CLSS.

6.46. Although the response is not exponentially growing, this limit cycle response is unstable for practical purposes. Because the linear dynamics of the bridge crane dominate its response, the experimentally determined critical gain of $K_o = 10$ is very close to the theoretical value of $K_o = 10.5$ predicted by the root locus.

A second set of experiments were conducted using different K_p and K_d gains in the controller block "C" defined by (6.15). The root locus of this system is shown in Figure 6.47. In this case, the root locus predicts that $K_o = 12$ will result in instability. The 1 meter step responses shown in Figure 6.48 again demonstrate that $K_0 = 10$ results in a limit cycle response. However, this response is different from that shown in Figure 6.46, exhibiting a significantly higher amplitude.

These experimental results match fairly well with the theoretically predicted affects of utilizing input shapers within feedback loops. The difference between the predicted and actual critical value of K_o is mostly due to the real system's nonlinearities which are not accounted for in the linear model. However, it should be noted that the range of K_0 values experimentally tested on the HighBay crane was rather

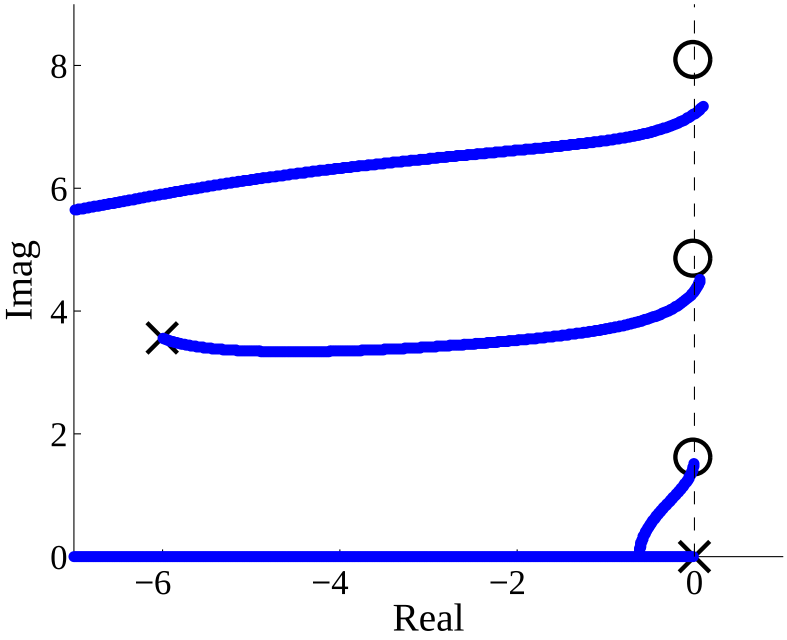


Figure 6.47: Root Locus of Second Crane Control System.

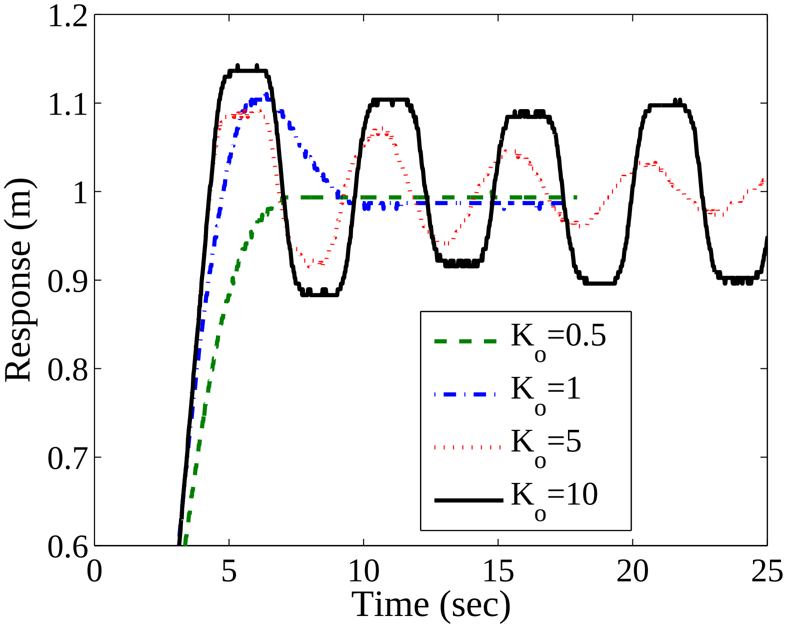


Figure 6.48: CLSS Crane Response with New PD Gains.

course. In fact, the only K_0 value tested between $K_0 = 5$ and $K_0 = 10$ was $K_0 = 7$. This value did not result in sustained oscillations for either set of K_p and K_d gains.

Therefore, the actual critical, K_0 value which results in instability (for both experiments) lies somewhere between seven and ten. However, the precise critical, K_0 value is not of critical importance, as the root locus prediction will always be wrong due to the nonlinear nature of the crane. The important point is that the root locus technique shown here gives a quick, reasonable estimate of the K_0 value which will result in instability. This will always need to be verified experimentally if the precise value is needed. Also, the root locus technique shown here gives the engineer a good understanding of the closed-loop system's secondary dynamics.

6.6 Stability Analysis Discussion

This chapter has shown how simple, standard stability analysis tools (the root locus and Bode plot) can be used to study the stability properties of CLSS controllers. Despite the fact that CLSS controllers add partial time delays to the feedback loop, these controllers can be stabilized even when some amount of modeling error occurs. However, it is clear from this chapter that CLSS controllers are more susceptible to instability than standard PID feedback control. The next chapter will demonstrate useful applications for CLSS controllers, where they give some advantage over PID control.

It is also important to note that this chapter only looked at two, basic plant types: a second-order plant and a fourth-order plant based upon a mass-spring-mass system. Using the techniques presented here, future work could address the stability of CLSS controllers that utilize more complicated plant types, including non-minimum phase systems.

CHAPTER VII

APPLICATIONS OF CLOSED-LOOP SIGNAL SHAPING CONTROLLERS

7.1 Introduction to CLSS Applications

The previous chapter established guidelines and boundaries for insuring BIBO stability of CLSS controllers. However, just because a control scheme is stable does not mean that it is useful. This chapter will investigate the usefulness of CLSS controllers for some typical control problems. CLSS will be compared to basic OLIS/PID combinations. This chapter will primarily use a second-order plant, but broader conclusions will be drawn from these results. The control issues studied here are:

1) Actuator/Force Disturbances

2) Sensor Disturbances

3) Modeling Errors

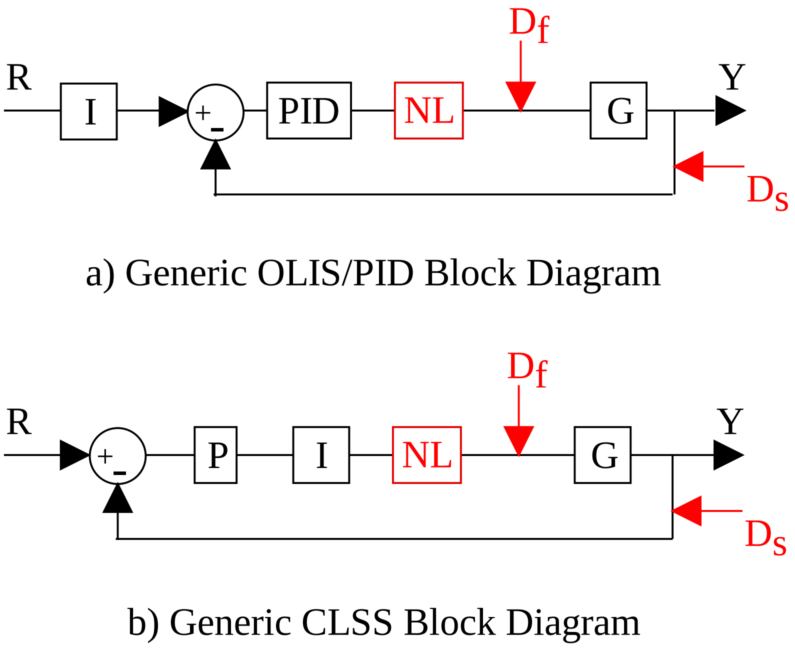
4) Trajectory Tracking

5) Non-collocated Control

6) Discontinuous (Hard) Nonlinearities

7) Improving the Performance of Human Operated Systems

Figure 7.1 shows the basic block diagrams that will be studied in this chapter. The top block diagram (Figure 7.1a) is an OLIS based controller. The bottom block diagram (Figure 7.1b) is a CLSS controller. These block diagrams also depict some of



a) Generic OLIS/PID Block Diagram

b) Generic CLSS Block Diagram

Figure 7.1: OLIS and CLSS Block Diagrams.

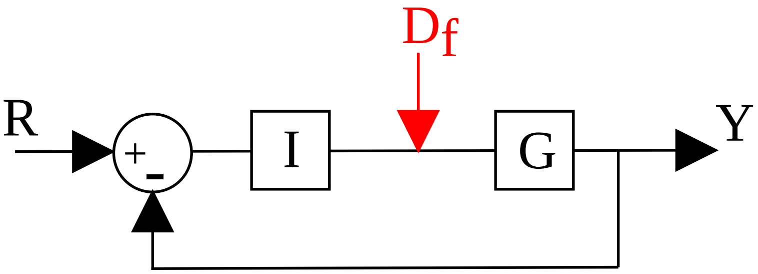


Figure 7.2: Block Diagram of Feedback System with Actuator Disturbances.

the control problems to be studied here: force disturbances (D_f), sensor disturbances (D_s) and system nonlinearities (NL).

7.2 Actuator Disturbance Rejection

7.2.1 Theoretical Perspective

Figure 7.2 shows a Classical Method CLSS control system that experiences actuator disturbances. Here, the P controller is set to unity and I represents the input shaper. The transfer function from R to Y is:

$$\frac{Y}{R} = \frac{IG}{1 + IG} \tag{7.1}$$

Under perfect modeling conditions, the oscillatory, open-loop poles of G are canceled by the open-loop zeros of I. Because the IG term appears in both the numerator and the denominator, evaluating the $\frac{Y}{R}$ transfer function at the open-loop poles of G would result in a finite value. This means that the open-loop poles of G are not closed-loop poles of the $\frac{Y}{R}$ transfer function.

However, this cancelation does not fully occur in the transfer function between D_f and Y:

$$\frac{Y}{D_f} = \frac{G}{1 + IG} \tag{7.2}$$

Because I appears in the denominator but not the numerator, the open-loop poles of G are also closed-loop poles of $\frac{Y}{D_f}$ (in addition to the closed-loop poles obtained from the characteristic equation $1 + IG = 0$). If the $\frac{Y}{D_f}$ transfer function is evaluated at the open-loop poles of G, the denominator would be finite, but the numerator would be infinite. Hence, just as in the case without feedback control, the system responds to an actuator disturbance according to the dynamics inherent to the plant, G. From these observations, it is not expected that this form of closed-loop signal shaping will provide any benefits over outside-the-loop input shaping combined with PID feedback control in terms of actuator disturbance rejection.

7.2.2 Second-Order Plant Simulation Results

Figure 7.3 shows the response to an actuator disturbance pulse (the reference input, R, was set to zero) for an OLIS/PD controller and a CLSS controller of the forms depicted in Figure 7.1. For this analysis, the plant was a damped, second-order system of the form:

$$G = \frac{1}{s^2 + 0.1s + 1} \tag{7.3}$$

Both input shapers were Zero-Vibration (ZV) shapers. The ZV shaper used in the OLIS/PD controller was tuned for the closed-loop poles created by the PD controller with $K_p = 1$ and $K_d = 0.4$. The ZV shaper used in the CLSS controller was tuned for

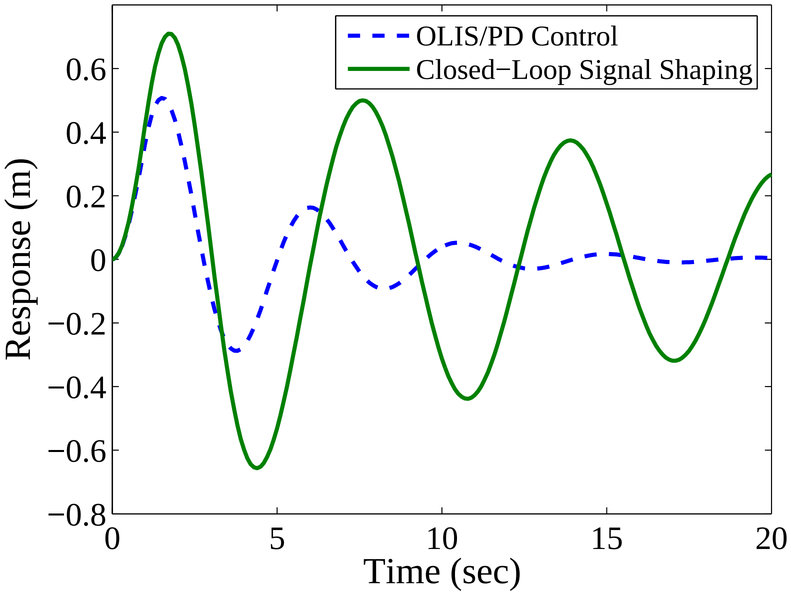


Figure 7.3: Actuator Disturbance Response of Second-Order Plant.

the poles of G. The simulation results shown in Figure 7.3 support the previous theoretical developments. It is clear from this figure that the closed-loop signal shaping scheme is significantly worse than the OLIS/PD controller. By utilizing derivative control, the OLIS/PD controller is able to create a closed-loop system with a high damping ratio. Since the CLSS controller responds to a force disturbance at the low-damped poles of G, the OLIS/PD controller has superior disturbance rejection. CLSS, in the form depicted in Figure 7.2, yields no advantage in terms of force disturbance rejection.

7.3 Sensor Disturbance Rejection

7.3.1 Theoretical Perspective

Figure 7.4 shows a CLSS controller that experiences sensor disturbances. As was done in the previous section, analyzing the output/disturbance transfer function will give some insight into the controller's ability to reject sensor disturbances. Fortunately, the transfer function between D_s and Y will not have closed-loop poles arising from

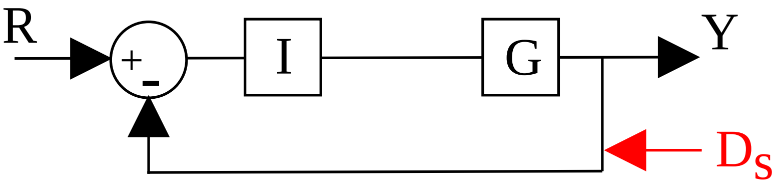


Figure 7.4: Basic Block Diagram of Feedback System with Sensor Disturbances.

the plant dynamics. The $\frac{Y}{D_s}$ transfer function is:

$$\frac{Y}{D_s} = \frac{-IG}{1+IG} \tag{7.4}$$

From the $\frac{Y}{D_s}$ transfer function, it can be seen that the open-loop poles of G are completely canceled by the open-loop zeros of I in both the numerator and denominator. Remember that this also occurs in the $\frac{Y}{R}$ transfer function. Just as in the $\frac{Y}{R}$ transfer function, the presence of I in the numerator and denominator means that the $\frac{Y}{D_s}$ transfer function will not have poles corresponding to the open-loop poles of G. The only poles associated with the $\frac{Y}{D_s}$ transfer function will come from the zeros of the characteristic equation $(1 + IG)$. However, these poles can be made much more damped than those of the lightly-damped plant being canceled by the input shaper. This ability to take a lightly-damped plant and create and significantly higher-damped closed-loop system without the need for derivative control shows that CLSS does have benefits over OLIS/PD control in regards to sensor disturbance rejection.

7.3.2 Second-Order Plant Simulation Results

The simulation results shown in Figure 7.5 support the theoretical developments of Section 7.3.1. Note that these simulations used the exact same plant, ZV shapers, and PD gains used in the actuator disturbance study. Figure 7.5 shows that both control schemes (CLSS and OLIS/PD control) are capable of achieving quick disturbance rejection. However, the OLIS/PD controller is only able to accomplish this because it uses derivative control. These simulations show that a CLSS controller can equal, or

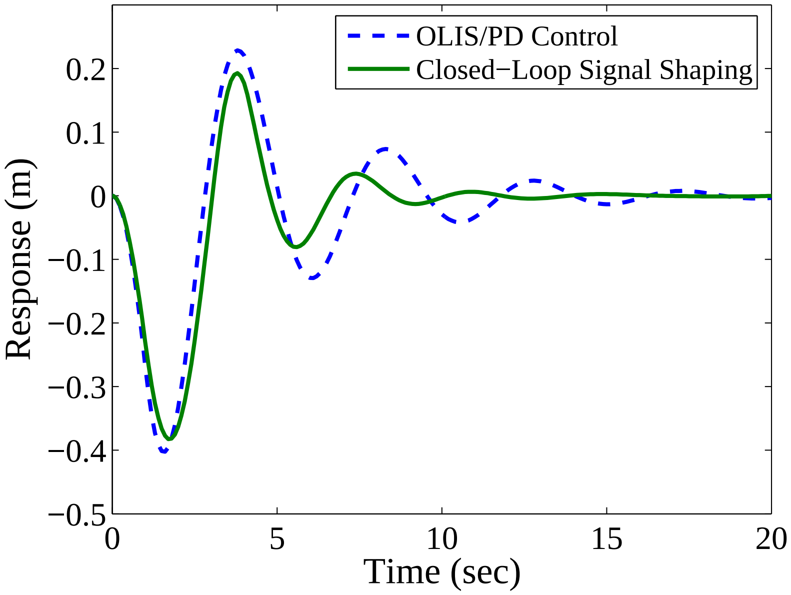


Figure 7.5: Sensor Disturbance Response of Second-Order Plant.

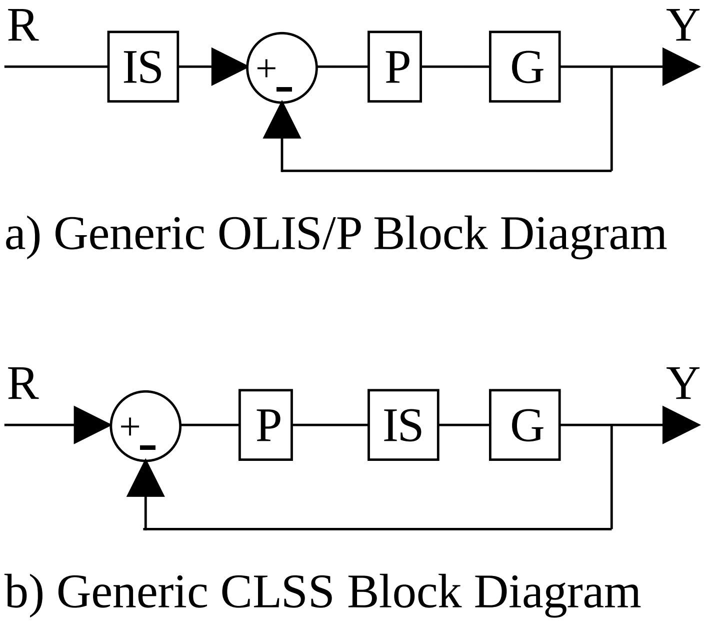
surpass, the performance of a standard PID controller in the area of sensor disturbance rejection without the need for differentiators.

It should be noted that the results shown in Figure 7.5 are for one specific plant and one set of PD gains. These results do not mean that OLIS/PD controllers cannot outperform (in terms of sensor disturbance rejection) CLSS controllers. The major result of this section is to show that CLSS controllers can adequately filter the response to sensor disturbances (creating a relatively high-damped response) without using derivative control. This ability to eliminate differentiators will make CLSS control a useful alternative to standard PD feedback controllers in practical situations where differentiators are difficult to implement.

7.4 Modeling Errors

7.4.1 Theoretical Perspective

Whether an input shaping filter is placed outside of the feedback loop, or inside it, the input shaper seeks to cancel some set of oscillatory poles. Modeling errors prevent the exact cancelation of oscillatory poles by an input shaper, resulting in some non-zero



a) Generic OLIS/P Block Diagram

b) Generic CLSS Block Diagram

Figure 7.6: Block Diagrams Used to Study Frequency Modeling Errors.

level of vibration. This section will investigate natural frequency modeling errors on second-order plants. This section will study the effects that these modeling errors have on the performance of OLIS/PD controllers and CLSS controllers.

Figure 7.6 shows the two controllers that are analyzed in this section. The controller shown in Figure 7.6a is a OLIS/P controller, whereas the controller shown in Figure 7.6b is a CLSS controller with proportional control. These block diagrams are limited to the configurations shown so that the only difference is the placement of the input shaping filter. Of course, this will also change how the input shaper is designed. When placed outside of the feedback loop, the shaper is tuned to the poles of the closed-loop system. When placed inside the loop, the shaper is tuned to the poles of the plant (G).

Figure 7.7 shows an open-loop zero of a ZV shaper (note that a ZV shaper places one zero at the location of the modeled dynamics) and the closed-loop pole of a feedback controller when there is no modeling error (left-hand side) and when some error in the prediction of the plant natural frequency does occur (middle and right).

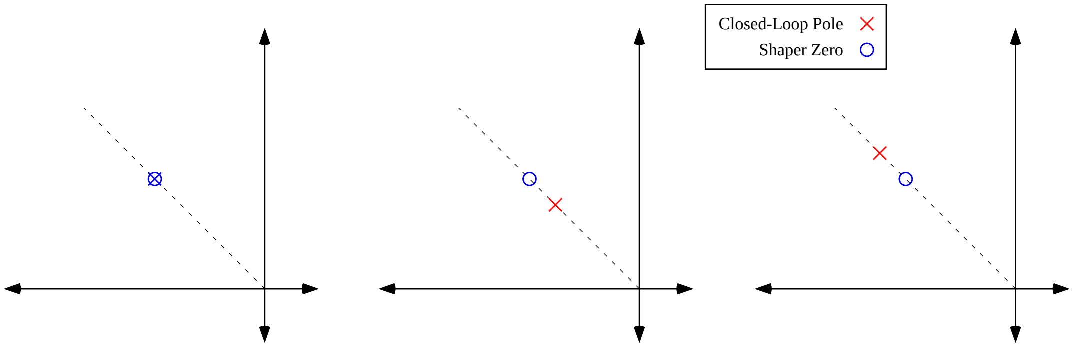


Figure 7.7: Pole/Zero Plot of Outside-The-Loop Input Shaping.

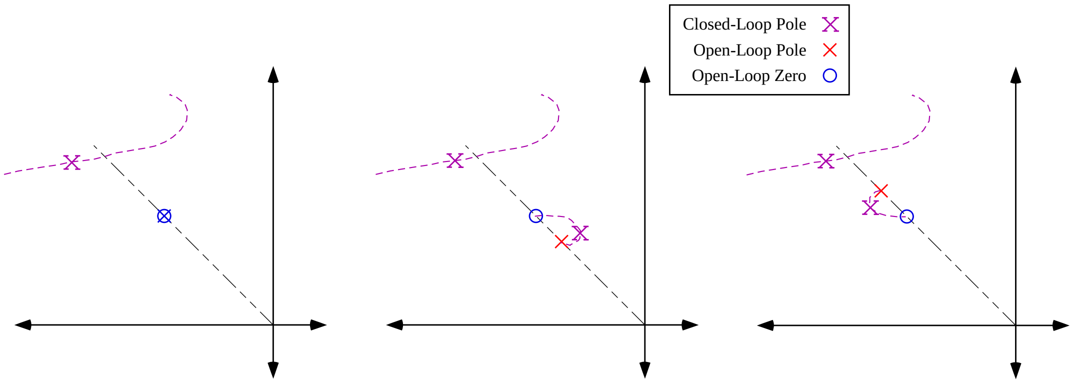


Figure 7.8: Root Locus of Closed-Loop Signal Shaping.

Note that this figure shows two plant poles for the modeling error case; one for actual frequency being lower than modeled (middle plot) and one for it being higher than modeled (right-hand plot). In outside-the-loop input shaping, even when some modeling error occurs, the poles remain relatively close to the shaper zero. The response of the poles is still somewhat attenuated by the proximity of the input shaper zero.

On the other hand, Figure 7.8 shows the root locus of a CLSS control scheme without modeling error (left) and with modeling error (middle and right). With the same modeling error as depicted in the open-loop case, closed-loop signal shaping creates two distinct possibilities. If $\omega_{act} < \omega_{mod}$ (middle plot), it is possible for the closed-loop pole to lie farther from the shaper zero and to the right of the original open-loop pole. Since the closed-loop pole is farther to the right than its open-loop counterpart, it has less damping and a longer settling time. This suggests that the closed-loop signal shaping response would be worse than the outside-the-loop input shaping response. However, if $\omega_{act} > \omega_{mod}$ (right plot), then the closed-loop pole can lie to the left of its outside-the-loop counterpart. Therefore, it has more damping and a quicker settling time. This suggests a superior response for closed-loop signal shaping. Unfortunately, modeling errors are usually not known (i.e. it is not usually known whether $\omega_{act} < \omega_{mod}$ or $\omega_{act} > \omega_{mod}$).

7.4.2 Second-Order Plant Simulation Results

A simple set of simulations performed using a second-order plant support the theoretical predictions given in the discussion above. The second-order plant in this section was again:

$$G = \frac{1}{s^2 + 0.1s + 1} \tag{7.5}$$

The shapers used in both controllers were ZV shapers. For this first set of simulations, the proportional gain was set to unity. The modeling error used here was 20%. That is, $\frac{\omega_{act}}{\omega_{mod}} = 0.8$ or 1.2.

Figure 7.9 shows that when $\omega_{act} < \omega_{mod}$, closed-loop signal shaping control fails to outperform OLIS/P control. However, when $\omega_{act} > \omega_{mod}$, Figure 7.10 shows that closed-loop signal shaping yields quicker settling times but also greater overshoot.

A more thorough comparison of these two control schemes compared the resulting overshoot and 2% settling time over a range of K_p values and modeling errors. The modeling error was accomplished by tuning the shapers to frequencies not equal to the actual plant or closed-loop frequencies. The modeled frequency used to tune the input shapers is labeled as ω_{mod}, whereas the actual frequency of the plant or closed-loop system is ω_{act}. The modeling error used in the following figures is represented as the ratio between modeled and actual frequency ($\frac{\omega_{act}}{\omega_{mod}}$). The overshoot and settling times are also displayed as ratios. The overshoot ratio is the maximum output reached

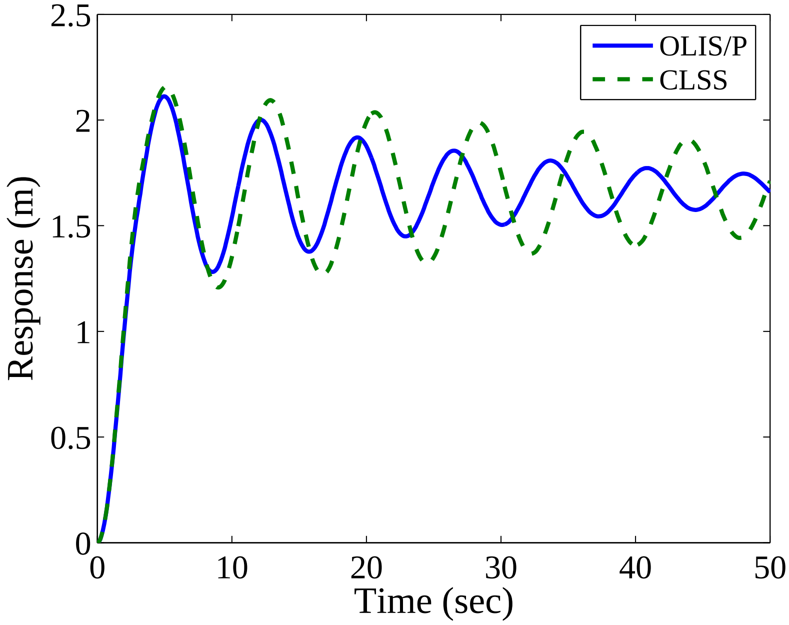


Figure 7.9: Modeling Error Response, $\omega_{act} < \omega_{mod}$.

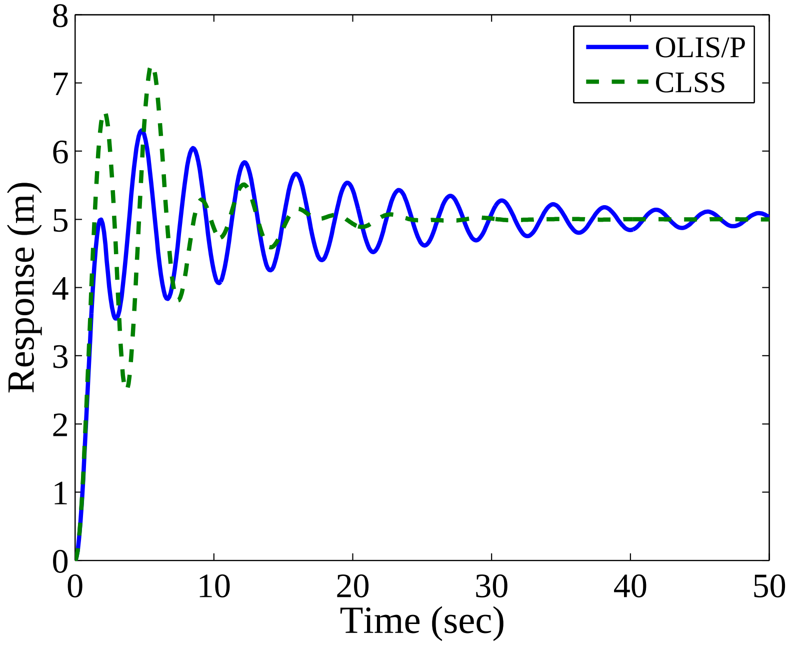


Figure 7.10: Modeling Error Response, $\omega_{act} > \omega_{mod}$.

by the OLIS/P controller divided by the maximum output reached by the CLSS controller. The settling time ratio is similar: settling time of the OLIS/P controller divided by the settling time of the CLSS controller. Since it is desirable for each performance measure (overshoot and settling time) to be small, a ratio less than unity

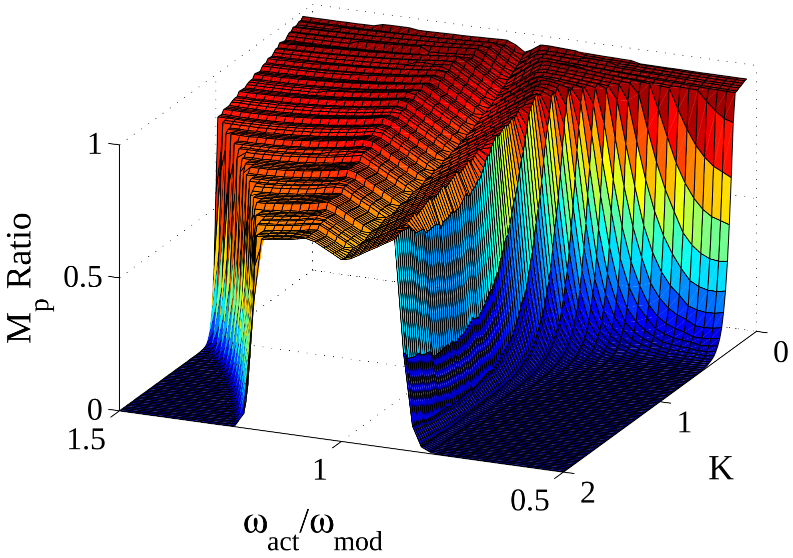


Figure 7.11: Overshoot Comparison when Modeling Errors Occur.

means that the OLIS/P controller performed better. A ratio above unity means that the CLSS controller performed better.

Figure 7.11 shows that the OLIS/P controller consistently outperformed the CLSS controller in terms of overshoot. The overshoot ratio is always less than, or equal to, unity. However, in terms of settling time, the CLSS controller was occasionally superior. This is depicted in Figure 7.12, where the settling time ratio is sometimes greater than unity. Note that for a significant set of modeling error/K combinations, the settling time ratio is equal to zero. This occurs when the modeling errors and/or the high proportional gains cause the CLSS controller to go unstable. This was discussed in detail in Chapter 6.

This section showed that CLSS controllers can have less settling time than OLIS/P controllers when modeling errors occur. However, this improvement in settling time is highly dependant upon the actual modeling error. For many modeling error values, the OLIS/P controller has a smaller settling time. In addition, Chapter 6 showed that CLSS controllers are more easily (as compared to a PD feedback controllers) driven unstable by modeling errors. Also, for the setup studied in this section, OLIS/P

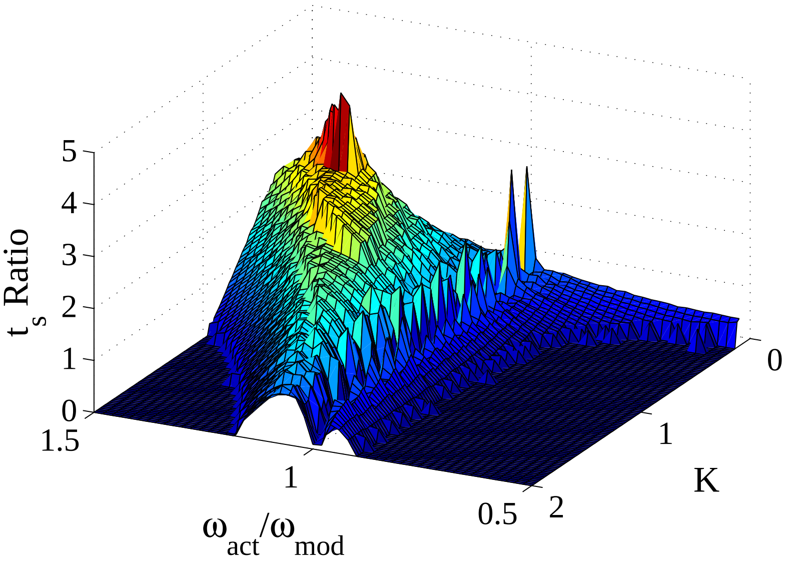


Figure 7.12: Settling Time Comparison when Modeling Errors Occur.

control consistently had less overshoot than CLSS control. Given that modeling errors are often unpredictable, this dissertation will advocate that CLSS controllers should not be used as a replacement for standard PID feedback controllers when plant modeling errors are of major concern.

7.5 Trajectory Tracking

One important area of controls research is the ability to track complicated trajectories. This section will compare the trajectory tracking ability of CLSS and OLIS/PD controllers. The trajectory used in this section is a circle.

The system studied here is the fourth-order system shown in Figure 7.13. This system is a damped mass-spring-mass system. For simplicity, the directions of motion and directional dynamics are assumed to be uncoupled. Also, the system is assumed to move only via translation in the X and Y directions. No rotation is allowed. The driven mass is M_1, and the mass required to follow the circle trajectory is M_2. Each direction has the same spring constant (k) and dashpot constant (b).

This system will first be controlled under the collocated scheme depicted in Figure

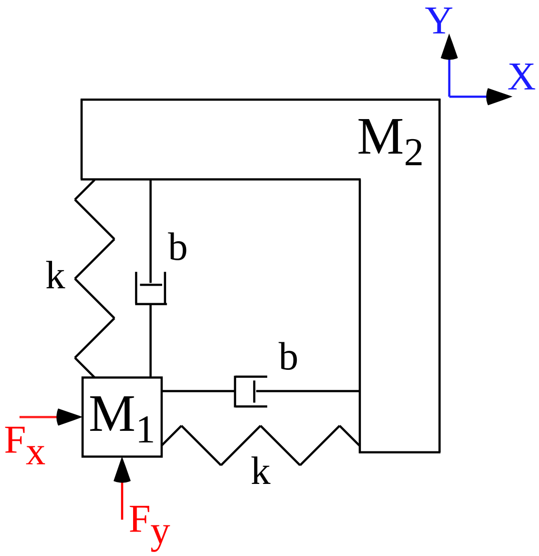


Figure 7.13: Fourth-Order, Mass-Spring-Mass System.

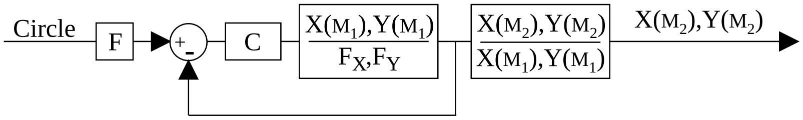


Figure 7.14: Block Diagram of Collocated, MSM System.

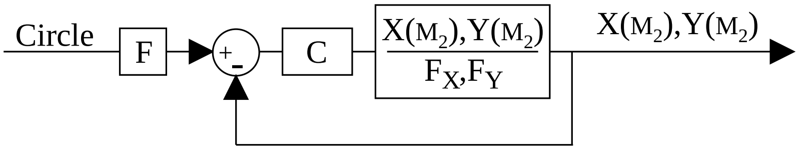


Figure 7.15: Block Diagram of Non-Collocated, MSM System.

7.14. Here, the actuators and sensors are located on M_1. Then, the non-collocated control scheme shown in Figure 7.15 will be analyzed. For the non-collocated case, the actuators remain at M_1, while the sensors are attached to M_2. These block diagrams depict the plant and both of the control schemes being compared. Here, the F block represents a feedforward filter which is an input shaper for the OLIS/PD controller and unity for the CLSS controller. The C block represents the feedback control block, which is a PD block for the OLIS/PD controller and an input shaper for the CLSS controller.

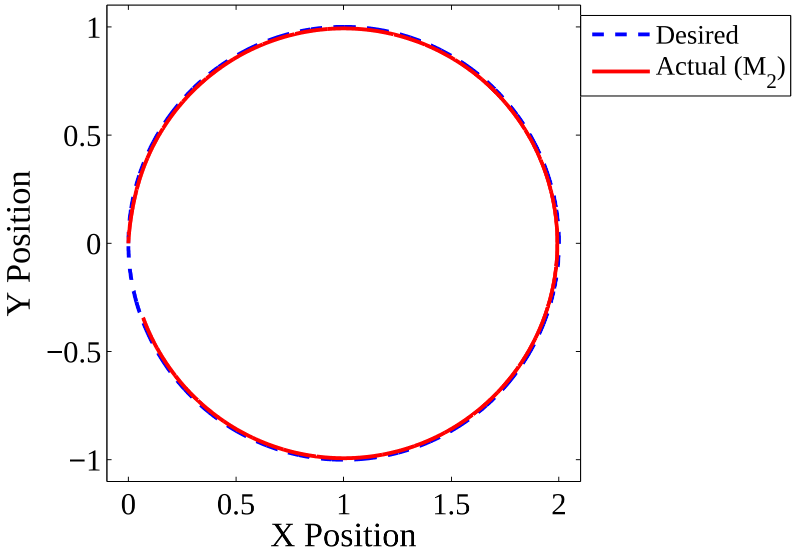


Figure 7.16: Collocated Trajectory Tracking - OLIS/P Controller.

7.5.1 Trajectory Following Under Collocated Control

Under collocated control, the position of the first mass is well controlled, while the second mass must remain under open-loop control. This is why PID control alone (no outside-the-loop input shaper) would be unsatisfactory. The motion of the second mass is not directly controllable via feedback, so the reference input must be filtered so that the second mass (M_2) does not vibrate.

For this collocated control study, OLIS/PD control is very capable of following the circle trajectory. This can be seen in in Figure 7.16, where the desired and actual responses of M_2 are shown. For these collocated simulations, $M_1 = 10$, $M_2 = 1$, $k = 1$ and $b = 0.1$. These gains were chosen to represent the motion of a main mass (M_1) that is attached to some smaller mass (M_2) via a connection with little stiffness and damping. This setup is a common problem in the field of motion control. One real-world example of such a system is spacecraft with light appendages.

Note that for this collocated setup, the OLIS/PD controller was actually capable of generating good trajectory tracking without any derivative action. For the response shown in Figure 7.16, the control scheme was actually a OLIS/P controller.

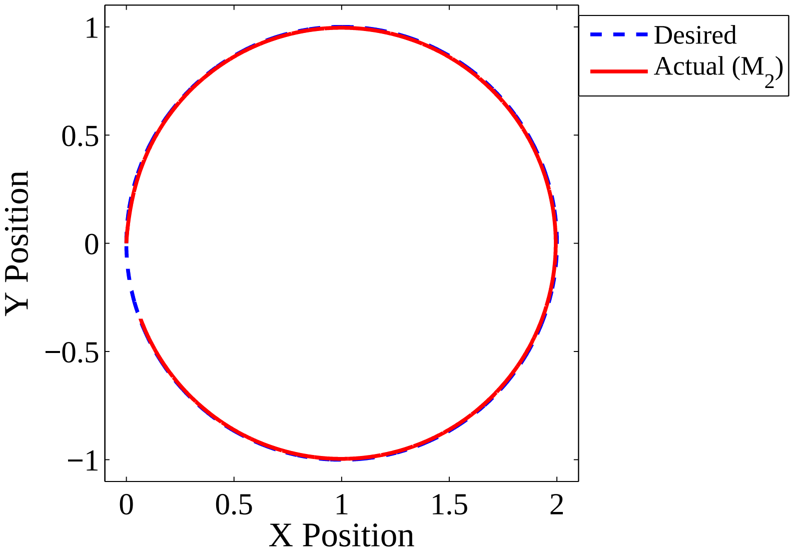


Figure 7.17: Collocated Trajectory Tracking - CLSS Controller.

Figure 7.17 shows that a CLSS controller can also be designed to have good trajectory tracking. Both the OLIS/P controller and the CLSS controller provide good tracking for the desired, circular motion. In addition, both controllers required reasonable and similar actuator effort profiles. Therefore, in the case of collocated control, CLSS has no obvious advantage over OLIS/P. One typical advantage previously assigned to CLSS was the ability to match the performance of an OLIS/PID controller without integrators or differentiators. However, for these collocated responses, the OLIS/P controller yielded good responses without integral or derivative action.

However, this collocated trajectory following example does provide a reasonable alternative to OLIS/P control. For example, Section 7.3 showed that CLSS controllers can reject sensor disturbances without the need for differentiators. Also, future sections will discuss the unique applicability of CLSS controllers to systems with actuator saturation. For certain situations, like the ones just mentioned, it will often be preferable to use a CLSS controller over an OLIS/PID type controller.

7.5.2 Trajectory Following Under Non-Collocated Control

If the performance requirements necessitate the use of non-collocated control, then closed-loop stability can quickly become a problem. The following section will highlight this issue.

7.5.2.1 Stability Comparison of Collocated and Noncollocated PD Controllers

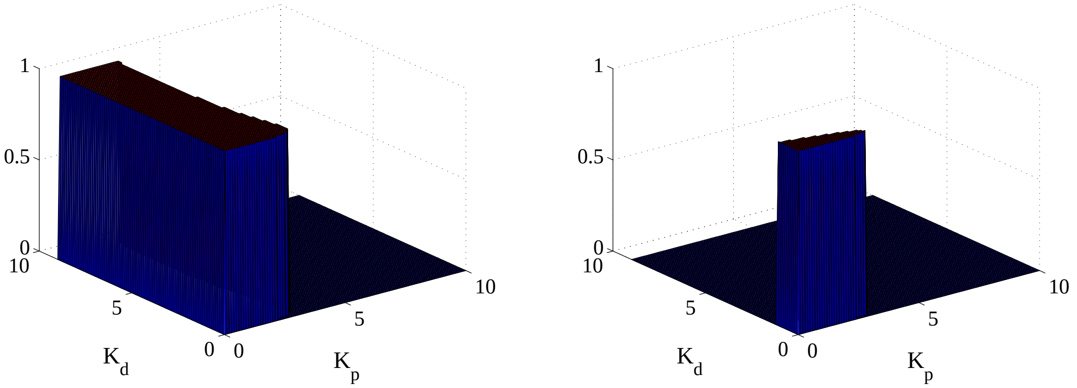
This section will analyze the stability of the control schemes shown in Figures 7.14 and 7.15. For this section, the controller (C) is assumed to be a PD controller. The closed-loop stability will be determined as a function of the PD gains. The influence of the spring constant (k) and dashpot constant (b) will also be analyzed. In the following figures, stable gain sets are indicated by a z-axis value of one, whereas unstable gain sets are indicated by a value of zero.

The first stability range comparison assumes the following system parameters:

$$M_1 = 10 \quad M_2 = 1 \quad k = 1 \quad b = 0.1 \tag{7.6}$$

Figure 7.18(a) shows the set of K_p and K_d gains that result in a stable closed-loop system for the collocated control scheme. Compared to Figure 7.18(b), which shows the stable gain sets for the non-collocated controller shown in Figure 7.15, collocated control enables a much larger set of gains. Note that if the dashpot constant (b) is reduced to zero, no set of PD gains will stabilize the non-collocated controller.

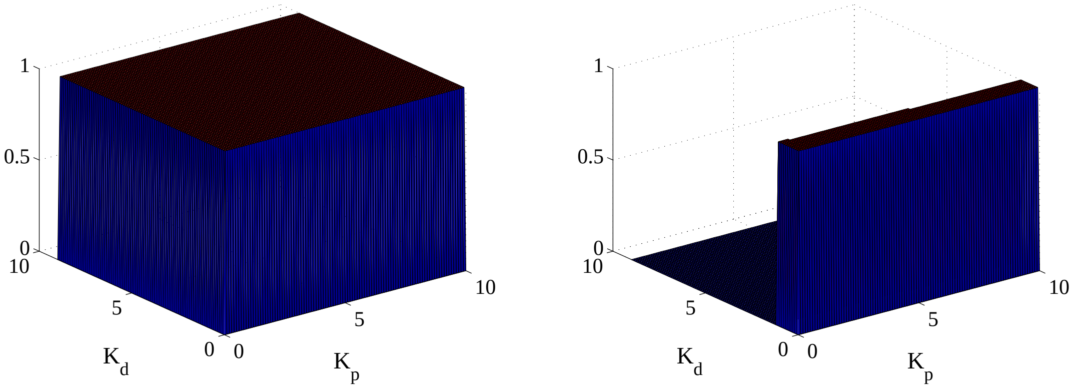
These gain sets can be enlarged by increasing the stiffness and dashpot constants. Figure 7.19(a) shows the increased gain set for the collocated case when the k value is increased from $k = 1$ to $k = 10$. Figure 7.19(b) shows the effect on the non-collocated controller's stable gain set with the higher k value. Figure 7.20(a) shows the stable gain set for the collocated controller when the k value is returned to unity but the b value is increased from $b = 0.1$ to $b = 0.3$. The stable gain set for the non-collocated controller with the higher b value is shown in Figure 7.20(b).



(a) Collocated Control

(b) Non-Collocated Control

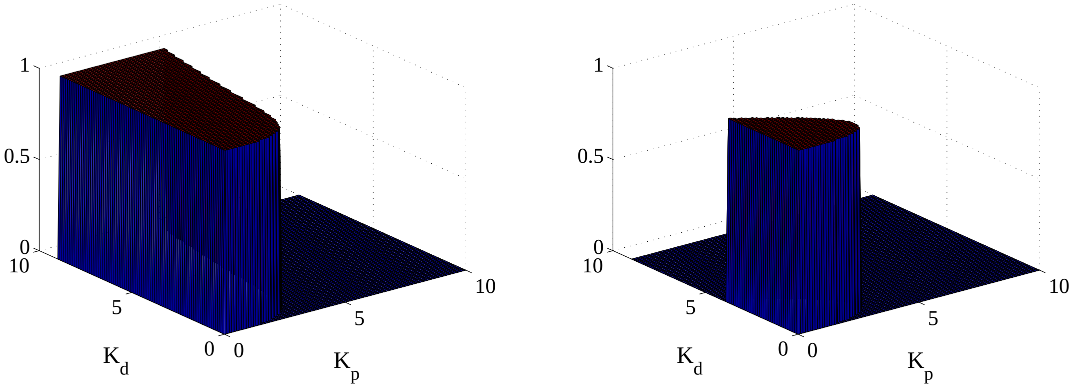
Figure 7.18: Stable Gain Sets for Low k and b Values.



(a) Collocated Control

(b) Non-Collocated Control

Figure 7.19: Stable Gain Sets with a Higher k Value.



(a) Collocated Control

(b) Non-Collocated Control

Figure 7.20: Stable Gain Sets with a Higher b Value.

There are two major conclusions from Figures 7.18(a) through 7.20(b). First, the non-collocated controller always has a significantly smaller set of gains (as compared to the collocated controller) that will yield a stable, closed-loop system. Secondly, as the stiffness and dashpot constants approach zero, the non-collocated controller becomes unstabilizable via PD control.

Because of the stability problems associated with non-collocated control, some recent research has begun to intentionally add time delays to non-collocated feedback systems as a means of ensuring stability [2,25,35,101]. As noted above, with the use of PD control alone, it can be quite difficult to achieve stability for non-collocated systems. In fact, Kumar reiterates the point that certain undamped systems under non-collocated control cannot be stabilized via standard PID control [35].

7.5.2.2 Effect of Stability Problems on Trajectory Following

The previous section illustrated that non-collocated control of a damped mass-spring-mass system is difficult to accomplish with PD control alone. Only a small set of PD gains will stabilize the system. The impact of this small gain set on trajectory following will now be examined.

One important aspect of using input shaping to aid trajectory tracking is that the trajectory has to be followed at a pace sufficiently slow in comparison to the duration of the longest input shaper being used in the controller [12,67]. Unfortunately, in order to achieve stability, the non-collocated, OLIS/PD controller is forced to have relatively small gains. This means that the oscillatory poles arising from the mass-spring-mass' rigid body mode have a relatively low frequency. The result is that the input shapers used for this lower mode are relatively long.

Assuming the system parameters ($M_1 = 10$, $M_2 = 1$, $k = 1$ and $b = 0.1$), the stabilizing set of PD gains is shown in Figure 7.18(b). This is a very small set of gains. A particular set of these stabilizing gains was chosen such that the closed-loop poles of the system had the highest possible natural frequency values. For the system analyzed here, a proportional gain of $K_p = 2.9$ and a derivative gain of $K_d = 0.3$ were chosen. The resulting two-mode closed-loop system had natural frequencies of $\omega_1 = 0.81\frac{rad}{sec}$ and $\omega_1 = 0.66\frac{rad}{sec}$. The two-mode ZV shaper used to filter the reference command to this OLIS/PD control system was tuned to cancel these frequencies.

The input shaper used in the non-collocated, CLSS controller is tuned to the plant frequency $\omega_1 = 1.05\frac{rad}{sec}$. This frequency is significantly higher than either of the frequencies shaped by the OLIS/PD controller.

The effect of the shaper lengths can be seen by requiring the two non-collocated controllers (CLSS and OLIS/PD) to follow a circular trajectory. If the circle is traversed slowly, the shaper lengths are short enough (compared to the speed at which the circle is traversed) that both control schemes enable good trajectory following. This can be seen in Figure 7.21.

The difference between the OLIS/PD controller and the CLSS controller becomes important if the damped, mass-spring-mass system is required to follow the circular trajectory at a higher speed. This is seen in Figure 7.22. Here, the CLSS controller

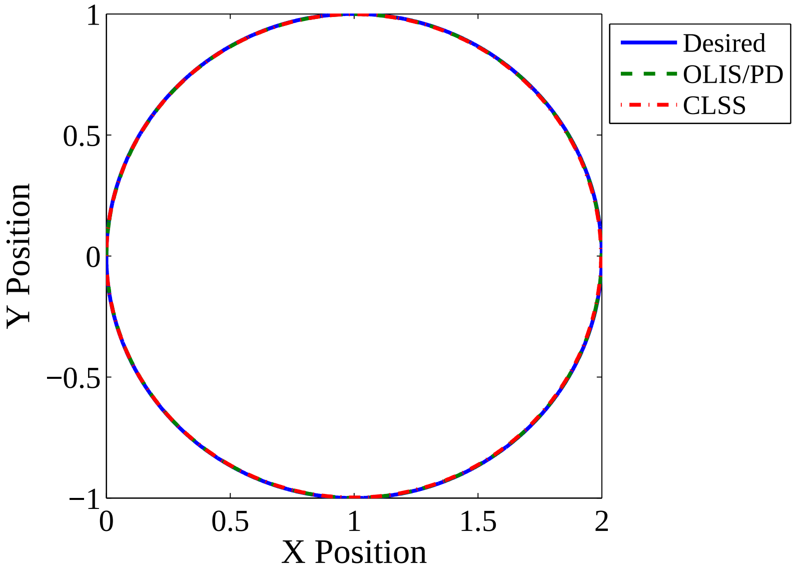


Figure 7.21: Trajectory Tracking Results Under Non-Collocated Control.

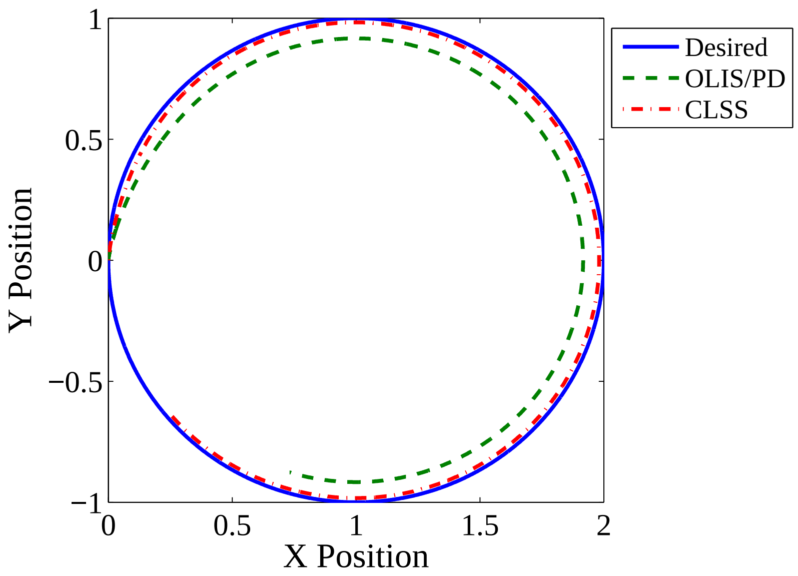


Figure 7.22: Fast Trajectory Tracking Results Under Non-Collocated Control.

provides significantly better tracking. Note that the time allowed for the circular trajectory was dropped from 200 seconds to 15 seconds.

The results presented in this non-collocated control comparison show that CLSS has another useful application. When PID based controllers have a restricted set of gains from which to choose, the overall performance of the control scheme can suffer. As shown in these examples, CLSS controllers have the ability not only to stabilize these non-collocated systems, but they can also yield better performance than OLIS/PD controllers. As mentioned before, the stability problems associated with non-collocated control have led researchers to intentionally place time delays inside of feedback loops. However, CLSS controllers have the added benefit that their specific combination of time delays is designed to cancel oscillatory dynamics.

7.6 System Nonlinearities

7.6.1 Actuator Saturation

One common nonlinearity seen in every real world application is actuator saturation. Controls engineers must use caution when pushing actuators beyond their rated limits,

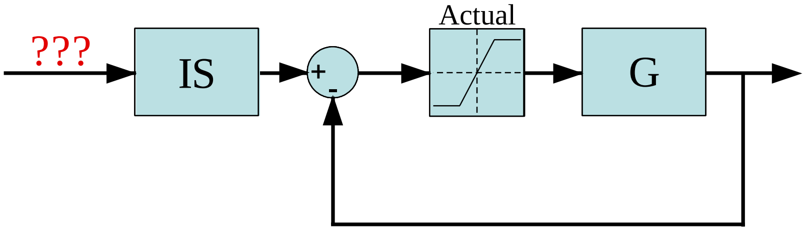


Figure 7.23: OLIS Control System with Saturation.

as this can cause complicated, unexpected and/or undesirable system behavior. In reality, many applications seek to avoid saturation completely so as to simplify the controller design and implementation. Input shapers are uniquely suited for systems with saturation issues. If the signal given to the shaper does not saturate the actuator, then the signal created by the shaper will also not result in saturation [1].

Input shaping is traditionally used outside of any feedback loops to filter a reference command. However, for many applications, the reference command has different units than the actuator input. For instance, a position reference command and force actuator input are of different units. In this case, if the input shaper acts upon the position reference command, it can be difficult to ensure that saturation will not occur. One would need to first design an unshaped reference input guaranteed not to result in actuator saturation. Then, the input shaper could be applied with confidence. This first step would often be a non-trivial task. A control scheme of this nature is depicted in Figure 7.23 with the unknown reference input labeled with question marks.

However, if the input shaper is used within the feedback loop and acts directly upon the force input to the actuator, the problem can be simplified. If a model of the saturation conditions the input to the input shaper, then the shaper will always produce a force signal attainable by the actuator. This input shaped force signal will also be completely filtered, so that unwanted frequencies are not excited. This kind of

[1] The only possible exceptions are input shapers containing negative impulses

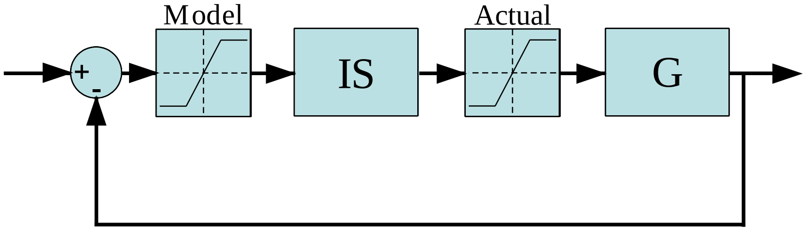


Figure 7.24: CLSS Control System with Saturation.

control system is shown in Figure 7.24. This very common issue of actuator saturation is one reason why a controls engineer might choose to place an input shaper inside the loop. In fact, as discussed next, this is one of the reasons why the Georgia Tech HighBay crane utilizes a CLSS controller.

7.6.2 HighBay Crane at Georgia Tech

The HighBay crane (located in the MaRC building) is a 10-ton bridge crane with at least four notable nonlinearities: a velocity limit, an acceleration limit, a built-in velocity smoothing algorithm that prevents sudden sign changes in velocity, and a velocity dead zone.

Khalid Sorensen has done an extensive amount of work on the Highbay crane [87–89]. To address the issue of actuator saturation, he chose the CLSS scheme shown in Figure 7.25. Here, "PD Control" is a Proportional-Derivative controller, the saturation block shown is a self-imposed actuator limit (like the saturation model depicted in Figure 7.24), "Drives & Motors" represents the nonlinear function which converts the commanded velocity of the overhead trolley to the actual velocity of the overhead trolley. This block contains all four nonlinearities mentioned in the

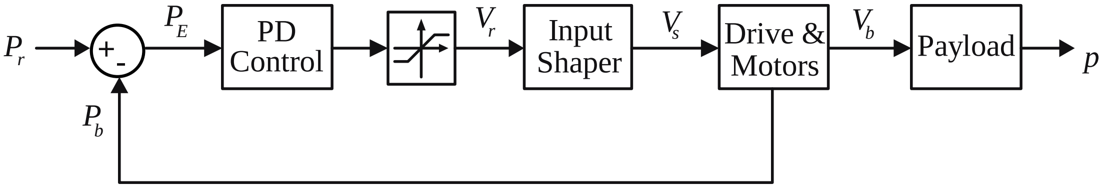


Figure 7.25: CLSS Control Scheme on Hibay Crane.

previous paragraph (which includes the actual velocity limit of the system). A final block, "payload" converts the overhead trolley's velocity into the resultant position of the crane's payload. The feedback control tracks the position of the overhead trolley, which is obtained via a laser range sensor.

For the same reasons mentioned in the previous subsection, Sorensen chose to handle the velocity limit (actuator saturation) by placing an input shaper inside the loop and filtering the shaper's incoming signal with a model of the crane's velocity limits. Note that the alternative would have been to place the input shaper outside the loop, filtering the position reference command. However, because of the velocity limit, this would require several extra steps to ensure that the position reference signal was of an acceptable form. This requirement is eliminated when the input shaper is placed inside the loop; any position reference command is acceptable.

The controller shown in Figure 7.25 was able to produce accurate and low-vibration motions of the HighBay crane. In fact, Sorensen conducted 25 randomly chosen point-to-point motion trials using the crane controller depicted in Figure 7.25 [87]. Figure 7.26 shows the final positioning error of the crane's trolley after each of these 25 point-to-point motions was completed. Figure 7.27 shows the vibration of the crane's

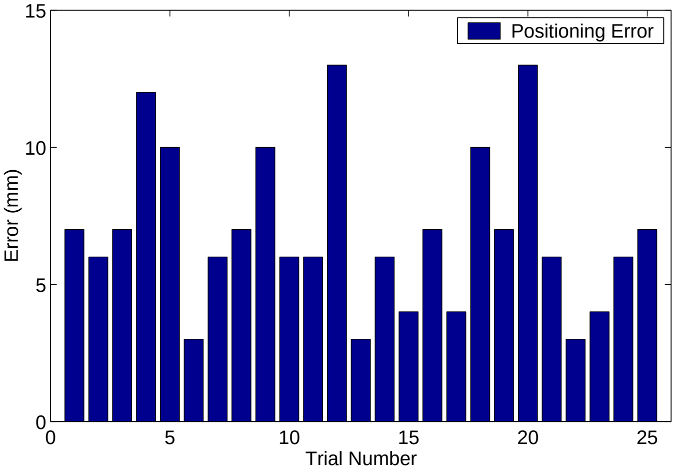


Figure 7.26: Trolley Final Positioning Error.

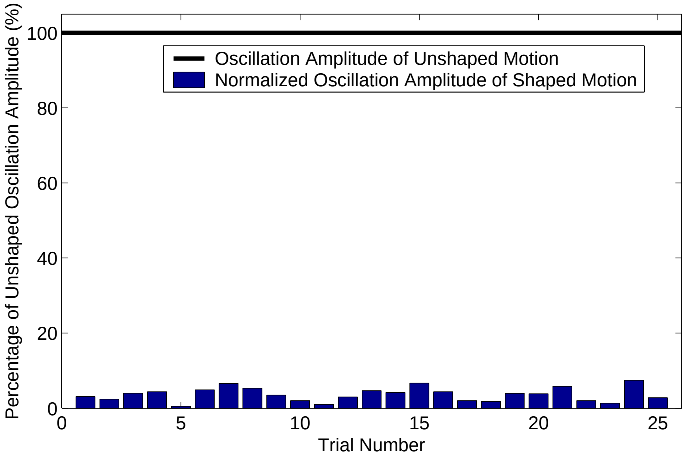


Figure 7.27: Residual Payload Oscillation.

payload upon completion of these point-to-point motions. Note that the oscillation amplitude is normalized by the amount of oscillation that normally occurs when the crane is driven without input shaping and under open-loop control (no PD control on the trolley). Both of these figures show that the CLSS controller depicted in Figure 7.25 was able to maneuver the HighBay crane accurately and without significant, residual oscillations.

In addition to the saturation issue, there were two additional reasons to implement CLSS on the HighBay crane. First, some initial simulations indicated that placing the input shaper within the feedback loop resulted in a better handling of the remaining nonlinearities grouped into the “Drives & Motors” block. When the input shaper was simulated outside the loop (even when velocity saturation was not an issue), this still resulted in poorer performance as compared to the CLSS controller. Secondly, Sorensen added a secondary PD feedback loop for the purpose of disturbance rejection. The full controller can be seen in Figure 7.28. The secondary PD feedback loop easily rejected disturbances. However, it would also move the crane from its desired position. By including the input shaper inside the “Input Shaping & Positioning Loop” feedback loop, the controller was able to use PD control to eliminate disturbances and PD control plus input shaping to maintain desired position.

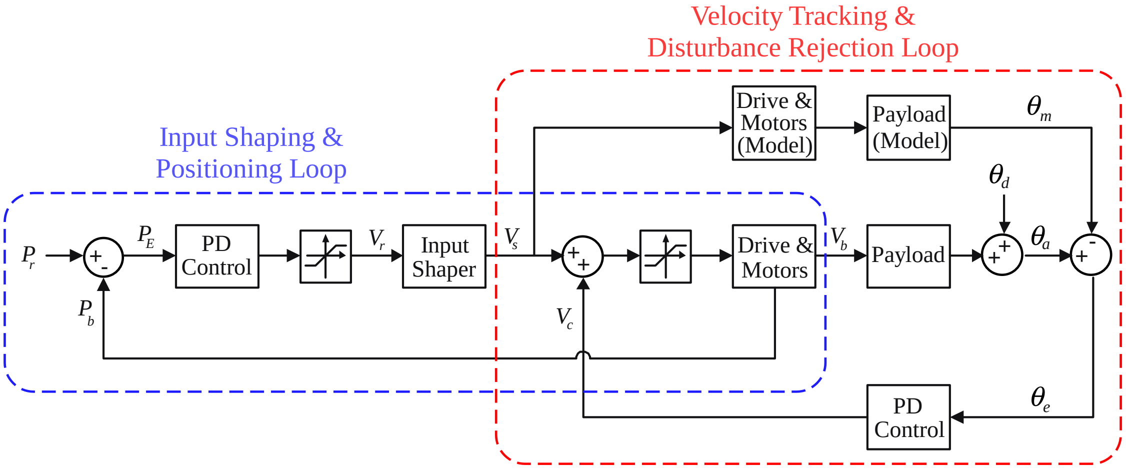


Figure 7.28: Full CLSS Control Scheme on Hibay Crane.

The HighBay crane is a real-world system that successfully uses a CLSS controller. The use of input shaping within the crane controller's feedback loop addresses several important issues inherent to this mechanism. Its successful usage on a real system is an important verification that CLSS has a reasonable realm of applicability.

7.7 Improving the Performance of Human Operated Systems

The last CLSS application studied in this dissertation is the field of human operated systems. Despite the ubiquitous presence of computer controlled systems, there are still many applications where human beings control the motion of complicated machinery; the need for human operated systems is not likely to disappear in the near future.

One system almost always controlled by a human in some manner is the crane. Cranes are an indispensable part of many industries, including oil drilling and exploration, mining, manufacturing, shipping and construction of all kinds. For these reasons, the study of human operated cranes is an important research task.

Some recent research in this area has focused on human operator performance on cranes equipped with input shaping [32, 33, 37]. The goal of this section is to further

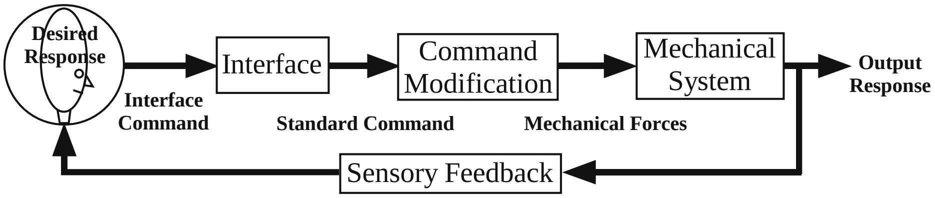


Figure 7.29: Human-Operator/Closed-Loop-Signal-Shaping Control Scheme.

those previous studies by evaluating whether or not the presence of input shaping technology enables human operators to maneuver cranes in a safer and faster manner. Typically, the control scheme being studied is similar to the one shown in Figure 7.29. Here, the human operator is attempting to complete some pre-determined task, such as to maneuver the crane's payload through an obstacle course as quickly as possible while attempting to avoid collisions with obstacles. The human operator commands the crane motion via an interface while also using his/her eyes to sense the actual motion of the crane and its payload. Based upon this sensory information, the operator adjusts the commands to the interface in real time. This is the type of controller that will be used throughout the experiments detailed in the section.

However, because the crane is an oscillatory system, an input shaper is often included within the controller to modify the human operator's commands. This typically eliminates the majority of the crane's oscillations. This placement of the input-shaping filter within the human-operator-feedback loop constitutes a form of closed-loop signal shaping.

Having justified the need to use human-centered CLSS controllers, the rest of this section will detail some research studying the performance of human operators maneuvering flexible cranes. The control schemes studied in this research are the CLSS controller shown in Figure 7.29 and an unshaped version of this control scheme. In this unshaped version, the input shaper is removed (made unity), forcing the human operator to perform all tasks associated with trajectory generation and error elimination. The main goal of this section is to compare the human operator's performance within these two controllers. The human operators were tested on a bridge crane and a tower crane. They were also tested under local control and remote control over the internet.

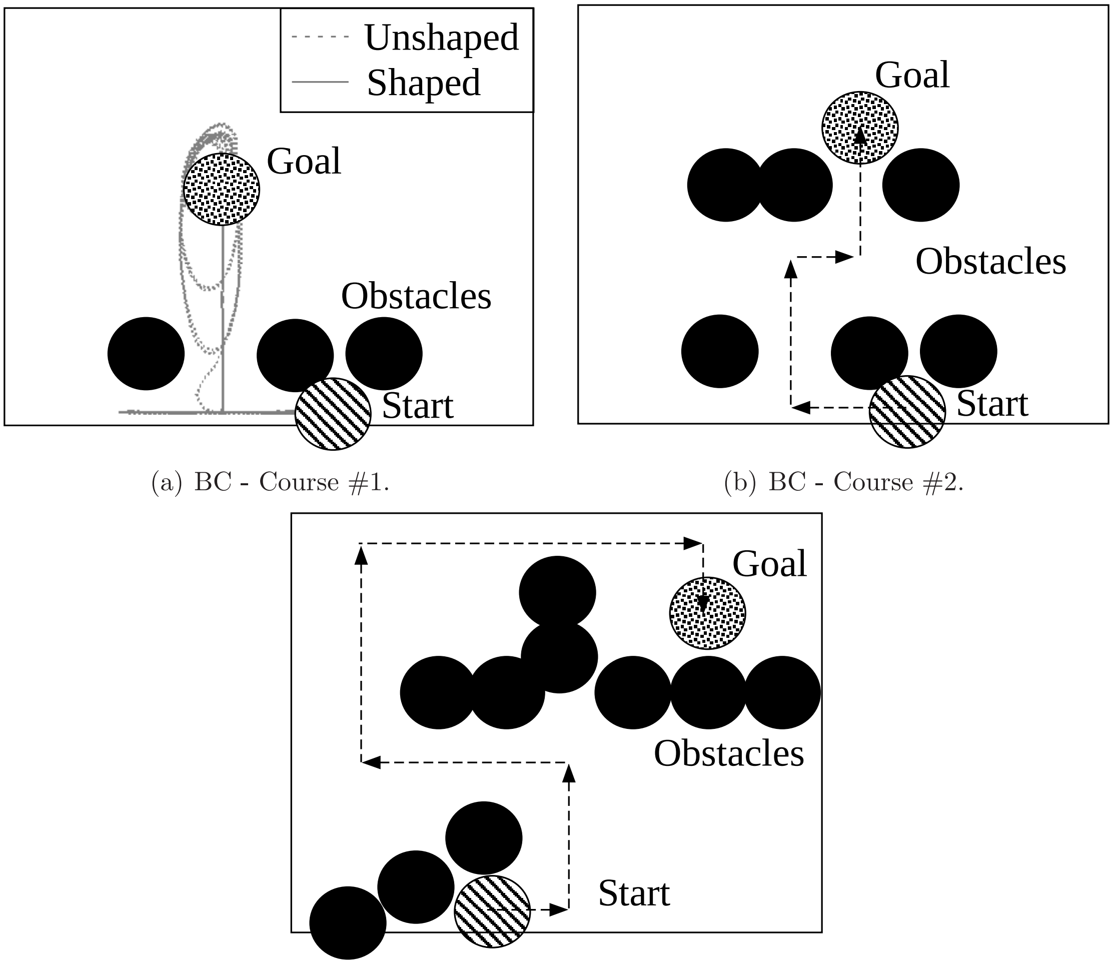
This human-operator study can be split into two sections. The first stage compared local and remote operation. The second stage observed the effect of task difficulty and complexity on the performance of a crane operator. Volunteer operators drove both a bridge crane and a tower crane through remote and local obstacle courses [37, 39]. The operator's performance was analyzed using several different obstacle courses. Performance was measured by run time (i.e. how long it took the operator to maneuver the crane through an obstacle field from a starting location to a finish point) and the number of collisions the crane's end-effector had with the obstacle field. For each stage of the this study, the effect of input shaping within the loop was analyzed. The experimental results indicate that remote manipulation of a crane via the internet is very challenging. However, the difficulty of the task is greatly reduced when closed-loop signal shaping is utilized.

7.7.1 Local vs. Remote Operation

7.7.1.1 Procedure - Bridge Crane in Atlanta, GA

An overhead view of the bridge crane obstacle courses at Georgia Tech can be seen in Figure 7.30. For the "Local vs. Remote" study, only Course #3 was used. The other courses were used to study task difficulty. Three volunteers operated the crane remotely and locally. In each case, the crane was operated with and without input shaping enabled. All tests were done twice, yielding a total of 24 runs. The operators were told that fast times and collision avoidance were equally important. The input shaper used was a single mode zero vibration (ZV) shaper [65, 85].

For local operation, the crane was controlled using a graphical user interface (GUI) on a computer directly connected to the crane. The GUI, similar to the one shown in Figures 7.31 and 7.32, allows the crane to be moved at full speed in four different



(a) BC - Course #1.

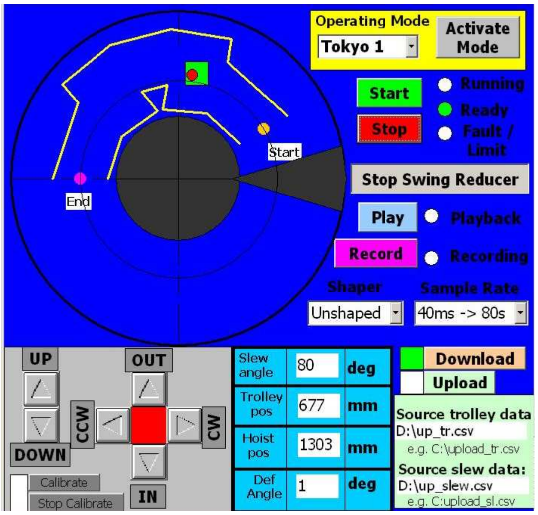
(b) BC - Course #2.

(c) BC - Course #3.

Figure 7.30: Bridge Crane Courses in Atlanta, GA.

directions (forward, reverse, left and right). It also provides a plot showing the layout of the course, the position of the crane trolley, and the position of the payload. To reduce the difference between remote and local operation, the operators looked at this GUI plot while locally controlling the crane instead of looking at the actual obstacle course.

For remote operation, the same GUI was displayed on a computer in another room of the same building using a protocol called VNC. However, under remote control, the crane's trolley and payload position had to be sent to the operator's GUI over the internet. This introduced an approximately one second delay in the feedback line.


Operating Mode Tokyo 1

Shaper Unshaped

Sample Rate 40ms -> 80s

Slew angle 80 deg

Trolley pos 677 mm

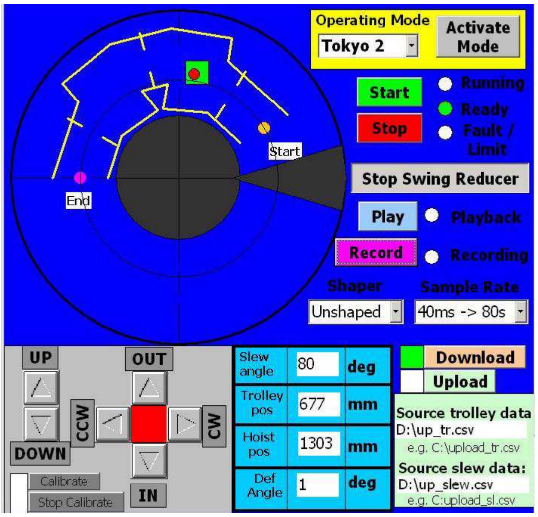
Hoist pos 1303 mm

Def Angle 1 deg

Source trolley data D:\up_tr.csv e.g. C:\upload_tr.csv

Source slew data: D:\up_slew.csv e.g. C:upload_sl.csv

Figure 7.31: "Tokyo 1" Course.


Operating Mode Tokyo 2

Shaper Unshaped

Sample Rate 40ms -> 80s

Slew angle 80 deg

Trolley pos 677 mm

Hoist pos 1303 mm

Def Angle 1 deg

Source trolley data D:\up_tr.csv e.g. C:\upload_tr.csv

Source slew data: D:\up_slew.csv e.g. C:upload_sl.csv

Figure 7.32: "Tokyo 2" Course.

7.7.1.2 Procedure - Tower Crane in Tokyo, Japan

Experiments conducted on a tower crane at Tokyo Tech utilized a setup similar to that described for the bridge crane at Georgia Tech. The GUI shown in Figures 7.31 and 7.32 was used for both local and remote operation. Figures 7.31 and 7.32 also depict the two obstacle courses used for testing on this tower crane. Note that the right-hand course is more difficult than the left-hand course. It has extra obstacles forcing the payload to move through narrow passages. The comparison between local and remote operation was done only for the most difficult course. Several people with varying levels of expertise operated the crane both locally and/or remotely. The remote runs were performed both within Japan and from other countries such as the USA. The operators experienced various levels of delay depending on their remote location.

7.7.1.3 Results - Bridge Crane in Atlanta, GA

Figure 7.33 shows the effect of remote operation and input shaping on run time and obstacle collisions. Remote operation obviously leads to increased run times, with an average increase of 24 seconds (75%) without input shaping and 13 seconds (60%)

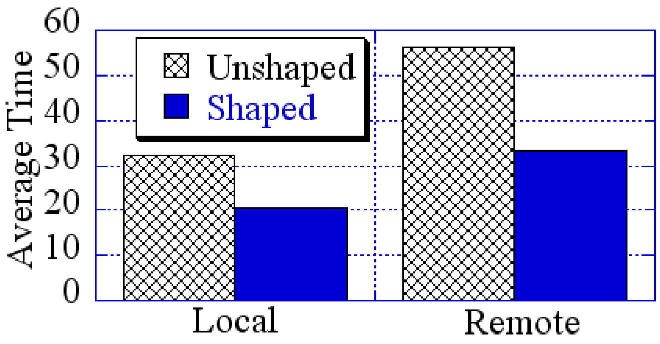


Figure 7.33: Bridge Crane Study - Run Times.

with input shaping. The use of input shaping leads to substantial time savings in both remote and local operation. Figure 30(a) depicts the advantage gained with input shaping by plotting an unshaped and a shaped run. Clearly, the presence of input shaping allows for faster and safer maneuvering by eliminating vibrations.

One of the problems that led to longer run times for remote operation was failure to compensate for the communication delay. This is a problem that could be somewhat corrected by improved operator training. A second, more difficult problem was the uncertainty created by the variable time delay. Because there was a variable delay in the transmission of commands and reception of visual feedback, it was difficult for the operators to execute precise commands. When attempting to make fine adjustments, the same length button press would sometimes result in too large of a movement and other times in no movement at all. Thus, instead of being to able to use intuitive knowledge of the crane dynamics, the operators had to give a command and then wait to make sure it was executed as expected. In some cases, the operator needed to make five or six adjustments before getting the crane to the correct position. When operating locally, such adjustments could usually be made with a single command. Input shaping is not particularly effective at solving this problem, so it is expected that run time would degrade when moving from local to remote operation, even with input shaping enabled.

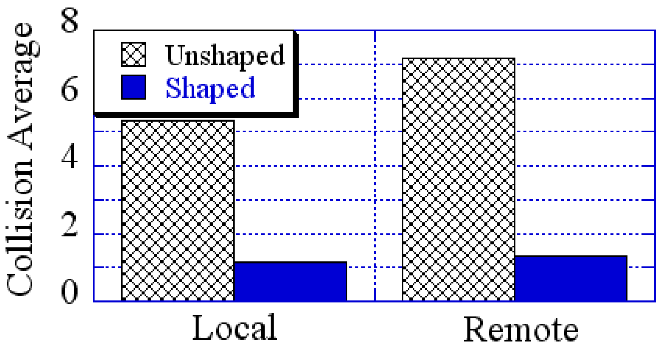


Figure 7.34: Bridge Crane Study - Collisions.

Figure 7.34 shows the average number of collisions that occurred. The most important result is that input shaping nearly eliminated collisions with obstacles. Without input shaping, remote operation increases the average number of collisions by 2.1 (40%), while it only increases the average by 0.1 (10%) with input shaping enabled.

The decrease in collision count when moving from remote to local operation without input shaping is explained by the ability of the operator to quickly take corrective actions. Most of the collisions were caused by payload oscillation. With local operation, the operator had enough feedback to move the crane in such a way as to perform some form of manual swing cancelation. Because of the long and varying time delay, remote operation made such maneuvers nearly impossible.

The relatively small improvement in collision count when moving from remote to local operation with input shaping enabled has a similar explanation. Because the primary cause of collisions was payload oscillation, the ability of input shaping to nearly eliminate oscillation solved the collision problem in both the remote and local runs.

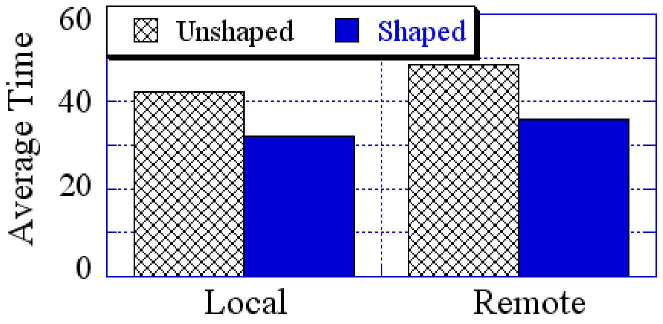


Figure 7.35: Tower Crane Study - Run Times.

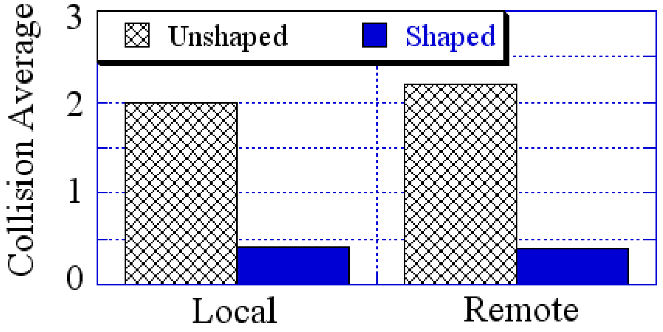


Figure 7.36: Tower Crane Study - Collisions.

7.7.1.4 Results - Tower Crane in Tokyo, Japan

Figure 7.35 shows the run time values for the tower crane at Tokyo Tech. Remote operation increased the run time by approximately 15%. The use of input shaping (the CLSS controller) reduced run times for both local and remote operation. Figure 7.36 shows the average number of collisions. As seen in these results, the average number of collisions without input shaping slightly increases when the crane is run remotely, but the same does not occur with input shaping. This confirms the results obtained from the experiments conducted in Atlanta, indicating that when input shaping is applied to remote operation the number of collisions is not significantly increased compared to local operation. Note that there is some uncertainty in the number of collisions. When the payload barely touches an obstacle, it is difficult to state definitively that a collision occurred. The number of collisions was determined by analyzing the recorded trajectory of the crane payload, as well as the X and Y velocity components, to detect the irregularities in the trajectory when the payload collided with physical obstacles. Only collisions that induced a noticeable change in velocity were counted.

7.7.2 Effect of Course Difficulty on Crane Operation

7.7.2.1 Procedure - Bridge Crane in Atlanta, GA

The second part of this study sought to determine the effect of course difficulty on the operation of cranes. Using the three different obstacle courses shown in Figure 7.30, each operator performed two runs with input shaping disabled and two runs with it enabled. All of the runs were performed using the same remote setup used in the "Local vs. Remote" tests described in the previous section.

The courses shown in Figure 7.30 are named after the number of turns (direction changes) required to complete the course. The arrows shown in Figures 7.30(b) and 7.30(c) (and the input shaped path shown in Figure 7.30(a)) indicate the general routes the operators were expected to follow. Each course was designed to have a different level of difficulty. Course difficulty was defined here as the number of turns an obstacle course had.

Generally, areas of direction change are when the most damaging oscillation can occur. If the crane trolley is stopped in one direction and ready to begin driving in a perpendicular direction, any payload vibration remaining in the initial trolley direction will endanger any attempted passes through narrow openings in the new trolley direction. Therefore, courses with more turns are likely to be more difficult; there are more chances for residual vibration to cause collisions (or to at least require a significant wait time for vibration to subside).

7.7.2.2 Procedure - Tower Crane in Tokyo, Japan

The tower crane studies also compared operator performance as a function of course difficulty. However, these trials focused on local operation, whereas the bridge crane tests in Atlanta focused on remote operation. For the tower crane operation study, course difficulty was precisely defined. The difficulty level D was defined to be the ratio of the payload suspension length, h, and the width of the path between the obstacles, w. This gives a quantitative level of difficulty, based on how much a payload can swing without colliding with an obstacle:

$$D = \frac{h}{w} \tag{7.7}$$

A larger value of D indicates a more difficult course. The easiest level would be when the path width is equal to, or larger than, twice the payload suspension length. In this case, the payload could swing freely, since the path is wide enough for any swinging amplitude.

As the level D becomes larger, the course difficulty increases, since the payload's swinging amplitude must be contained so as not to collide with obstacles. The level of difficulty can be increased until the width of path is equal to the payload diameter, $w_{payload}$. Therefore, the maximum difficulty level (for a given length, h) can be defined as $D_{max} = \frac{h}{w_{payload}}$.

After defining difficulty, two rigid obstacle courses were constructed with the same length and the same number of turns. Only the width of the path was changed. These two courses were shown in Figures 7.31 and 7.32 and are named "Tokyo 1" (left-hand figure) and "Tokyo 2" (right-hand figure). Simple examination of Figure 7.31 shows that the "Tokyo 1" course is easier than the "Tokyo 2" course. This conclusion is confirmed by the difficulty level values given here: $D = 3.7$ for "Tokyo 1" and $D = 8.7$ for "Tokyo 2". These courses were designed to be run with a payload height of $h = 1300mm$, and a payload diameter $w_{payload} = 90mm$. The course "Tokyo 1"

Table 7.1: Tokyo Obstacle Course Parameters

Course	Difficulty Level	Maximum Difficulty
Tokyo 1	3.714	14.444
Tokyo 2	8.667	14.444

has a path width of $w = 350mm$ and "Tokyo 2" has $w = 150mm$. These parameters are shown in Table 7.1.

7.7.2.3 Results - Bridge Crane in Atlanta, Ga

The results showed exactly what one would expect; the more turns a course had, the longer it took to navigate. However, this result is somewhat unfairly skewed towards the expected results.

As can be seen from Figure 7.30, the nominal path from start to finish increases with course difficulty. This means that it should take longer to complete Course 5 than Course 1. Therefore, in an attempt to fairly compare run times for each course, the completion time was normalized by subtracting the optimal time inherent to each individual course. The optimal time was calculated by first establishing a nominal path through each course, similar to the paths shown in Figures 7.30(b) and 7.30(c). The nominal paths were characterized by simple straight line motions roughly centered within the spacing between obstacles. The optimal time for each course was then calculated by adding the time required to traverse each of the independent straight lines at maximum velocity. The time wasted on a course was then obtained by subtracting the optimal time from the actual run time.

The average time wasted for each course is shown in Figure 7.37. With or without shaping, it is clear that increasing the number of turns increases the average time wasted. These results are expected, because before each turn the operator must ensure that they are properly positioned and try to minimize vibration. Additionally, it can be seen that input shaping greatly improves performance.

Figure 7.38 shows the average number of collisions that occurred. As one would

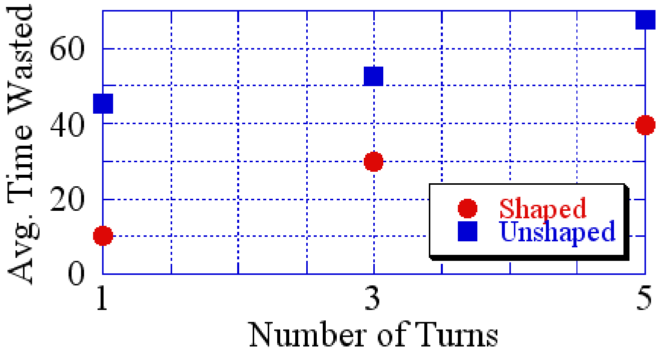


Figure 7.37: Bridge Crane Study - Time Wasted.

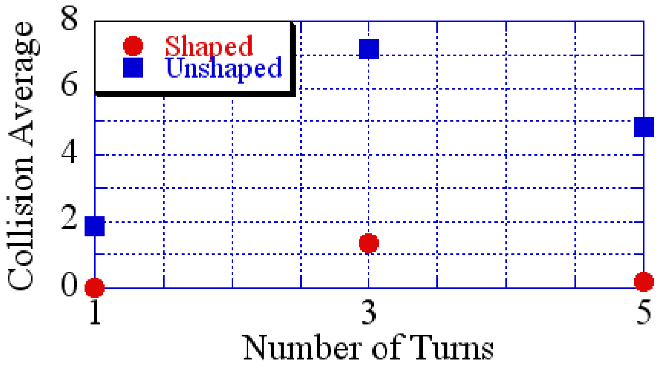


Figure 7.38: Bridge Crane Study - Collisions.

expect, the easiest course had the fewest number of collisions, and input shaping substantially reduces collisions.

However, the results for Course 5 seem counter-intuitive. For both operation modes, Course 3 resulted in more collisions than Course 5. The most likely explanation for this phenomenon is that Course 5 is a different kind of course. Most of the turns in Course 3 were tight, with many obstacles completely surrounding them and allowing little room for error. This can be seen in Figure 7.30(b). On the other hand, many of the turns in Course 5 were delimited by obstacles on one side, and a workspace limit on the other side. As a result, the payload was significantly less likely to collide with anything, as exceeding the workspace limits was not considered

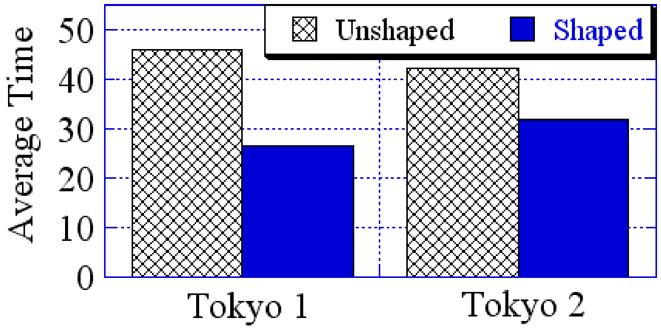


Figure 7.39: Tower Crane Study - Run Times.

a collision. This is the dominating effect that caused Course 5 to have fewer collisions than Course 3.

Another possible contributing factor for this is learning. Each of the three operators ran the courses in the same order: Course 3, Course 1, Course 5. Since Course 3 was run first, it was run with the most inexperienced operators. However, by the time Course 5 was run, each of the operators had already maneuvered the crane at least eight times.

7.7.2.4 Results - Tower Crane in Tokyo, Japan

Figure 7.39 shows the run time results for the tower crane. The results indicate that the unshaped run time of the difficult course is shorter than for the easy course. This is somewhat counterintuitive, and can be explained by learning, similar to the Atlanta results on collisions (Figure 7.38). Each operator began on the easy course, acquiring skills before proceeding to the more difficult course. However, regardless of learning, the presence of input shaping (using the CLSS controller) again enabled quicker obstacle course completion.

Figure 7.40 shows that the average number of collisions increases when the difficulty level is increased. However, when input shaping is used, the collisions are significantly reduced, once again confirming the benefit of input shaping.

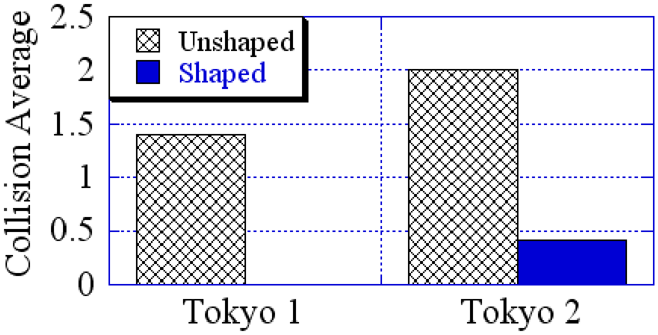


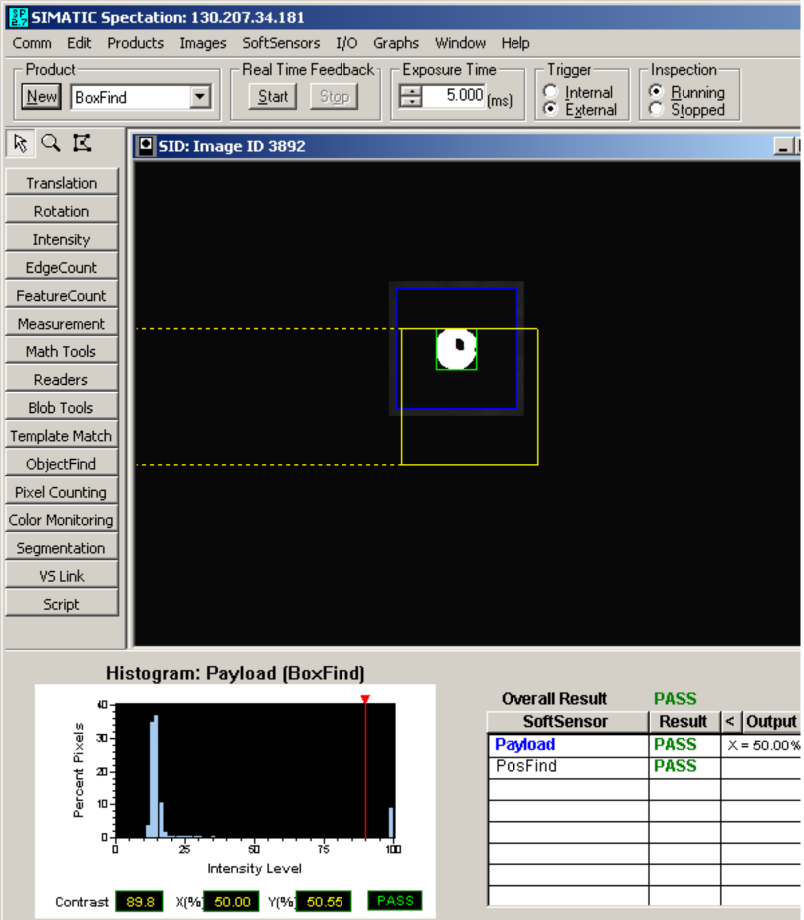
Figure 7.40: Tower Crane Study - Collisions.

7.7.3 Data Acquisition for Human Operator Studies

The operator performance data reported throughout Section 7.7 was acquired in two different ways. Some of the "Average Time" and "Collision Average" data were determined manually by timing the study participants and counting the number of collisions. This was the data collection process primarily used on the Atlanta based bridge crane. For the Tokyo based Tower Crane, run time and collision data were obtained from the recorded path of the crane's payload. Although not used to calculate run time and collision data, the payload paths were also recorded on the Atlanta based bridge crane. An example of recorded paths is shown in Figure 7.30(a).

Recording the actual paths taken by the human operators was important for several reasons. One, it enabled test subjects to operate the cranes without direct supervision. This meant that important performance measurements (run time and collisions) could be obtained after a test run was complete. This was especially important for the remote tests, where time zone differences made it difficult to have someone physically watching the crane's motion while it was being operated from a different continent. Second, the path information was critical to showing example motions, like the one shown in Figure 7.30(a). For the purpose of publishing this research, it was very important to have several example motions to highlight the various aspects of the human-operator study, like the differences between input shaped and unshaped motion.

In order to obtain this path information, a Siemens camera (Simatic VS 723-2) was used. This camera was able to distinguish the crane's payload from the background because the payload was fitted with retro-reflective tape. This made the light reflecting off the payload and entering the camera to be of a much higher intensity than light entering the camera from the background. A picture of the camera software program is shown in Figure 7.41. The white circle is the fiducial marker on the payload that is covered with retro-reflective tape. In the bottom left corner is a histogram showing the statistical information of light intensities entering the camera. The pixels with an intensity below a selected threshold are displayed as the color black. All of the high intensity information is the light from the payload. The pixels



SIMATIC Spectation: 130.207.34.181

Product: BoxFind

Real Time Feedback

Exposure Time: 5.000 (ms)

Trigger: ○ Internal ● External

Inspection: ● Running ○ Stopped

SID: Image ID 3892

Histogram: Payload (BoxFind)

Contrast 89.8 X(%) 50.00 Y(%) 50.55 PASS

Overall Result PASS

SoftSensor	Result	<	Output
Payload	PASS	X = 50.00%	
PosFind	PASS		

Figure 7.41: Data Acquisition Program.

corresponding to this high-intensity light are turned white. The result is a clear contrast between the payload (the white circle) and the background (black).

The research conducted for this thesis made two major advances within the program used to record the motion of the payload. The first was to create a small, moving digitization window. The camera can digitize an area up to 1024 pixels by 768 pixels. However, this requires a significant amount of processing time, increasing the sample time of the camera. Since the program in this study only needs to follow a small, circular object, the program was designed to only digitize a small box around the circle. This is shown by the blue square in Figure 7.41. By digitizing only this small number of pixels, the camera's sampling time was greatly reduced. Secondly, the camera initially had problems when the payload swing took the fiducial marker outside of the camera's field of view. This event would often cause the camera to fault and shut down. Because these cameras were being used to record a crane's motion while the crane was being remotely operated, this sort of camera failure would shut down all experiments until a person with personal access to the crane could restart the camera. This was an unacceptable characteristic that would make the remote studies very difficult to execute. Therefore, the camera program was specifically designed so that a loss of the fiducial marker did not result in a camera shut down. Instead, the camera increased the digitization window, waited for the fiducial marker to re-enter its field of view, and then re-focused the digitization window and continued following the payload. These camera program improvements made the remote operator studies much more feasible to execute and allowed for relatively high sampling frequency.

7.7.4 Conclusions on Human Operator Studies

The study of human-operator performance on cranes, as detailed in this section, shows that including an input shaping filter within the human-centered feedback loop improved speed and safety. Because human-operated, flexible systems form an important subset of controls research, the improvement made by utilizing a CLSS controller demonstrates its usefulness on real-world systems.

7.8 Discussion of CLSS Applications

This chapter studied the usefulness of Classical Method CLSS controllers applied to some of the most typical controller design issues: actuator disturbances, sensor disturbances, modeling errors, trajectory tracking, non-collocated control, some discontinuous, or “hard”, nonlinearities, and human-operated flexible systems. Given that OLIS and PID feedback control form a powerful and often-used control strategy, the capabilities of CLSS were compared to those of OLIS combined with some form of PID control.

The Classical Method form of CLSS studied in this chapter was incapable of outperforming OLIS/PID control in the realm of actuator disturbances and basic trajectory tracking. In the area of plant modeling errors, CLSS was sometimes (when the modeled frequency was lower than the actual frequency) superior to OLIS/PID control in terms of settling time. However, in terms of overshoot, the OLIS/PID controller was always superior to CLSS for the cases studied. In addition, the previous chapter showed that CLSS controllers often have stability problems when significant modeling errors occur. Therefore, the results of this dissertation do not support the specific use of CLSS controllers for the purpose of providing good responses when unpredictable and significant modeling errors are expected.

Fortunately, the Classical Method form of CLSS did prove to be a good alternative to OLIS/PID control in the remaining areas studied:

1. Sensor Disturbance Rejection
2. Actuator Saturation
3. Non-Collocated Control
4. Improving Human-Operated Flexible Systems

In the case of sensor disturbance rejection, CLSS control was able to quickly eliminate disturbances without the need for differentiators. In the area of saturation, CLSS controllers provided an easy way to implement input shaping on closed-loop systems with actuator limits. In addition, this section reviewed the use of CLSS on a real, 10-ton bridge crane, including experimental results. These experiments highlighted the CLSS controller's ability to maneuver the crane accurately and without significant oscillations. Its successful use on a real-world system is evidence that it can, and should, be used under certain circumstances. When studying trajectory tracking, it was found that CLSS controllers, by the very act of adding time delays to the feedback loop, created a stable, non-collocated system not easily achievable through standard PD control. When the trajectory was traversed quickly, the CLSS controller was able to provide better trajectory following than the OLIS/PD controller. Finally, the use of CLSS was discussed on human-operated cranes. It was shown that when the input shaper is included within the human-centered feedback loop, the operator was generally able to perform faster and safer maneuvers. This was true even if the operator was remotely controlling the crane over long distances that caused additional, visual-feedback delays.

CHAPTER VIII

ADVANCED CLOSED-LOOP SIGNAL SHAPING CONTROLLERS FOR FORCE DISTURBANCE REJECTION

Section 7.2 showed that the Classical Method form of closed-loop signal shaping controller, as depicted in Figure 7.2, fails to reject force disturbances. Fortunately, some interesting research has begun to address this problem by developing advanced forms of closed-loop signal shaping (CLSS) controllers [82, 84, 85, 91]. These new CLSS controllers: the Plant Inversion Method (PIM) and the Model Reference Controller (MRC), are specifically designed to reject force disturbances. This chapter will discuss both methods and show that each is left somewhat incomplete by the previous literature. These limitations will be rectified so as to produce useful and effective CLSS controllers for force disturbance rejection.

8.1 Plant Inversion Method Derivation

The first closed-loop signal shaping control scheme specifically designed for force disturbance rejection is based on a design originally presented by O.J.M. Smith [82,84, 85]. Smith began his work by stating that one would ideally like to create the control scheme shown in Figure 8.1. In this figure, C is some controller, G is the system plant, I_1 is the input shaper filtering the reference command R, and I_2 is the input shaper designed to filter the disturbance signal D. If possible, this control scheme would modify disturbances so that they do not excite the system's oscillatory modes. However, this control scheme is impossible to create under normal circumstances.

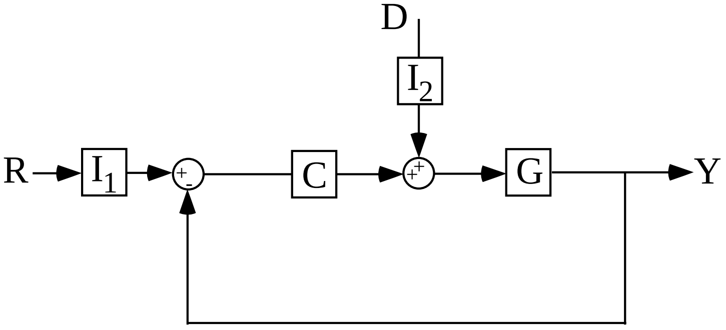


Figure 8.1: O.J.M. Smith's Desired Block Diagram.

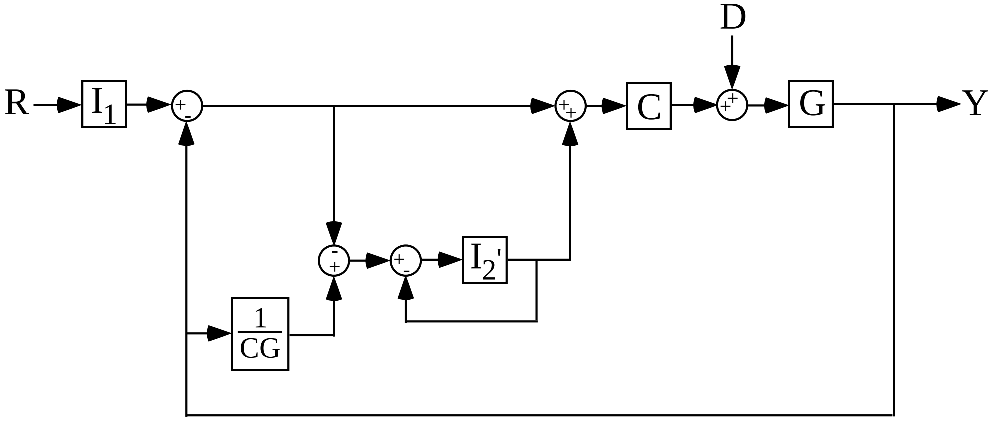


Figure 8.2: O.J.M. Smith's Original Concept.

The very nature of disturbance forces is that they usually act directly on the plant and are not alterable by the controller.

For these reasons, Smith designed the control shown in Figure 8.2 to mimic the behavior of the ideal control scheme shown in Figure 8.1. For this control scheme, I_2 is restricted to the class of input shapers whose first impulse is unity. That is:

$$I_2(s) = 1 + \sum_{i=2}^{n} A_i * e^{-st_i} \tag{8.1}$$

I_2' is simply the delayed portion of I_2:

$$I_2'(s) = \sum_{i=2}^{n} A_i * e^{-st_i} \tag{8.2}$$

This control scheme operates by using an inverted model of C and G to re-create the disturbance signal. The I_2' portion of the controller then adds additional force inputs to the plant so that the original disturbance signal is destructively interfered with and the system oscillations arising from the disturbance are canceled. In other words, the actual disturbance is treated as the first part of an input shaped signal. The inverted plant/controller model and partial input shaper (I_2') then create the remaining portions of the input shaped signal started by the actual disturbance.

Unfortunately, as shown in Figure 8.2, this scheme requires plant and controller inversion. One problem with this plant/controller inversion is causality. However, this problem has now been fixed by an alteration to the controller shown in Figure 8.2 that was developed in this dissertation.

To describe this alteration, the control system will be analyzed in the digital domain. The $\frac{1}{CG}$ block in Figure 8.2 is non-causal because the numerator has a higher-order z polynomial. This will always occur if the original plant (G) and controller (C) combination is strictly proper. However, remember that the I_2' block is solely comprised of time delays (z^{-m} terms, where $m > 1$). Therefore, this I_2' term can be rewritten as $z^{-n} * z^{n} * I_2'$. The z^{-n} term can then be removed from the inner I_2' feedback loop shown in Figure 8.2 and added to the $\frac{1}{CG}$ block. This will make the $\frac{1}{CG}$ block causal without making the $z^{n} * I_2'$ non-causal. The only condition is that n must be less than the smallest multiple of z^{-1} found in I_2' and greater than the polynomial-order difference making the inverted controller-plant block non-causal. This modified, PIM control scheme is shown in Figure 8.3. Note that several other z^{-n} terms are present to preserve the overall, functional equality between the PIM controllers depicted in Figures 8.2 and 8.3.

Often, causality is addressed by simply adding time delays to the control system. However, simply adding time delays to make the CG inversion causal, without adding the remaining delays and advances, results in unstable behavior for this system. However, because of the I_2' term, this control scheme is uniquely suited to fix this causality problem. The control has internal time delays that can be rearranged to ensure the causality of every block without the need for additional delays.

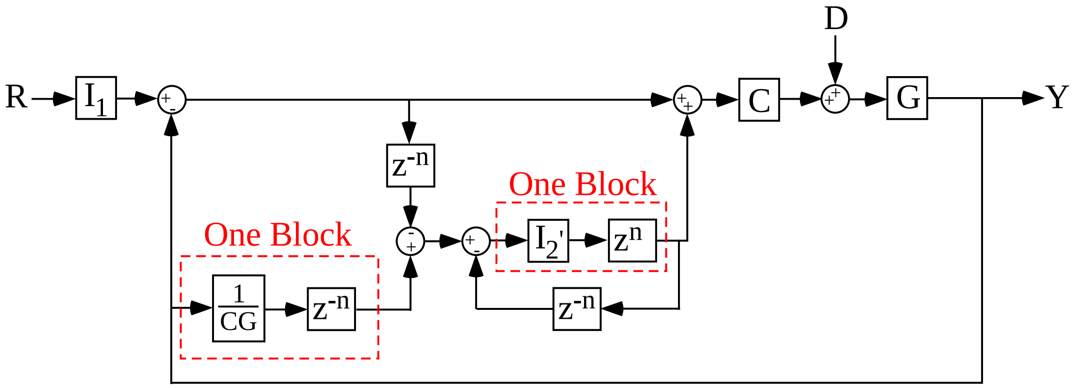


Figure 8.3: Modified, Causal Version of OJM Smith's Original Concept.

8.2 *Performance of Plant Inversion Method*

This section will investigate potential applications for the Plant Inversion Method. The issues addressed here include:

1. Step response performance - comparison to OLIS combined with PD control.
2. Effect of natural frequency modeling errors on step response.
3. Effect of sensor noise.
4. Ability of PIM to reduce disturbance-induced vibration in unobservable modes.

8.2.1 Step Response Performance

Figure 8.4 compares the unit-step and pulse-disturbance responses of a PIM controller and an OLIS/PD controller. The derivative action on the OLIS/PD controller was implemented via velocity feedback. The plant used here was a lightly damped second-order system. For the PIM controller, the block C was just a proportional controller ($C = K_p$). For the responses shown in Figure 8.4, $K_p = 1$ for both control schemes and K_d was varied from $K_d = 0.25$ to $K_d = 1$ in the OLIS/PD controller. The proportional gain has a similar effect on the PIM controller as it does on an OLIS/PID controller. Raising K_p increases the DC gain of the closed-loop system, as well as generally

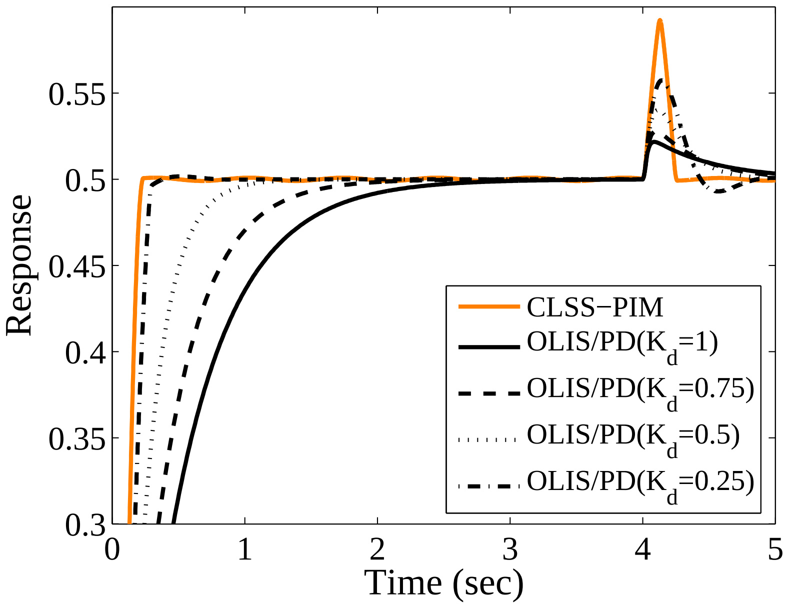


Figure 8.4: PIM vs. OLIS/PD Reference and Disturbance Response Comparison.

making the system more oscillatory. This translates into faster step responses when input shaping is used.

Figure 8.4 shows that the PIM controller has a faster step response and a faster disturbance rejection. However, the initial deviation caused by the disturbance is worse for the PIM controller. As the derivative gain is reduced, the OLIS/PD controller approaches the quick step response of the PIM controller, but also becomes more susceptible to disturbances. Figure 8.5 shows the actuator effort requirements for each controller. The main point in this figure is that neither control scheme requires a significantly more demanding actuator effort profile than is required by the other controller.

Figures 8.6 and 8.7 show the same comparisons under higher controller gains. The K_p gain in both the PIM and OLIS/PD controllers was increased to $K_p = 5$. Then, the derivative gain in the OLIS/PD controller was varied from $K_d = 0.5$ to $K_d = 3.75$. Again, Figure 8.6 shows that lowering the derivative gain causes the OLIS/PD controller to approach the performance of the PIM controller. However,

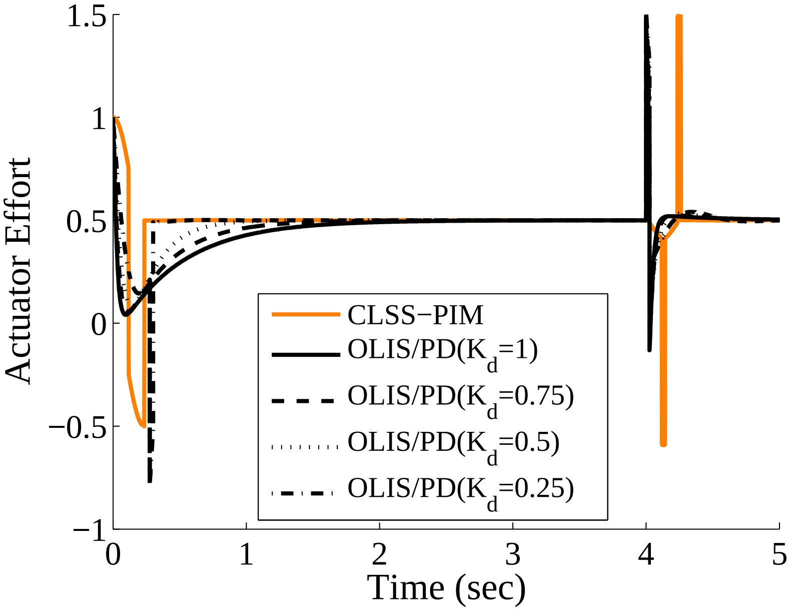


Figure 8.5: PIM vs. OLIS/PD Actuator Effort Comparison.

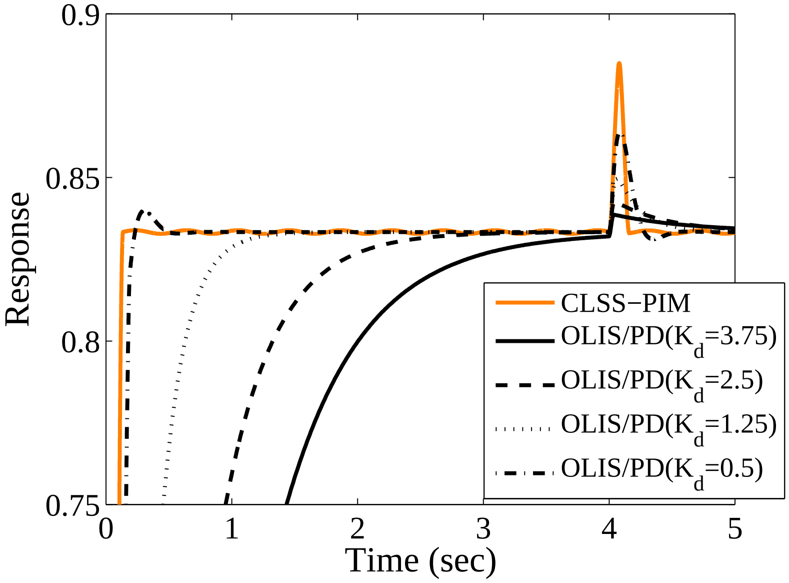


Figure 8.6: PIM vs. OLIS/PD Response Comparison - Higher K_p.

the PIM controller consistently has a faster step response and disturbance rejection. Finally, as shown in the previous example, Figure 8.7 verifies that both controllers require approximately the same actuator performance.

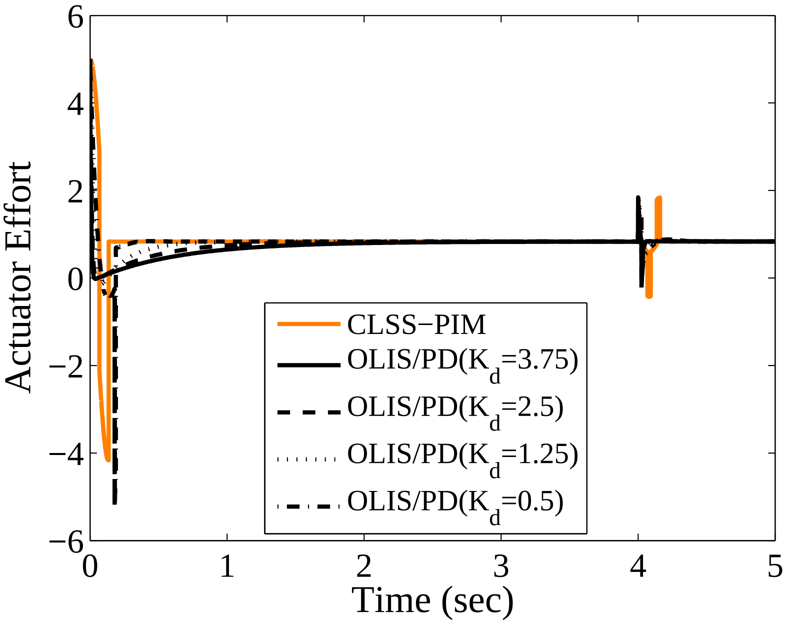


Figure 8.7: PIM vs. OLIS/PD Actuator Effort Comparison - Higher K_p.

These two control methods were also compared to lead compensator control. However, an OLIS/lead compensator control scheme could not be found that resulted in performance comparable to either the PIM or the OLIS/PD controller. The OLIS/lead compensator control schemes studied were significantly slower than either the PIM or OLIS/PD controllers. In addition, because of the closed-loop zero introduced by the lead compensator, the step reference response always experienced some amount of overshoot.

The major result derived from this section is that the PIM controller has a faster response than the OLIS/PD controller to reference inputs and disturbances. For cases where the speed of response and disturbance rejection are of primary concern, the Plant Inversion Method is a CLSS controller that provides a better response than standard OLIS/PD controllers. However, if maximum deviation caused by the disturbance must be limited, then the controller of choice would most likely be one that uses some form of PID feedback control.

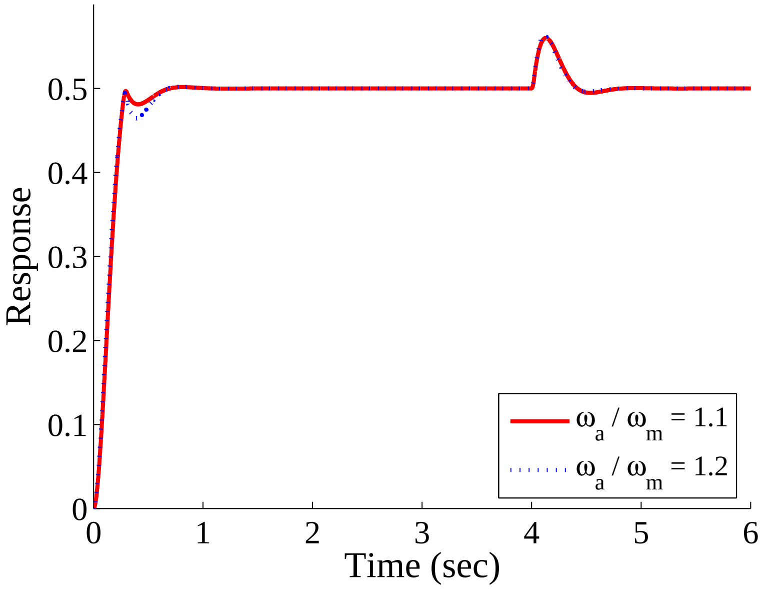


Figure 8.8: OLIS and Velocity Feedback, $\omega_a > \omega_m$.

8.2.2 Natural Frequency Modeling Errors

This section will compare the sensitivity of the PIM and OLIS/PD controllers to frequency modeling errors. Figure 8.8 shows the same OLIS/PD control scheme studied above with the actual plant natural frequency (ω_a) being larger than the modeled natural frequency (ω_m). It is clear from this figure that the system performance is slightly degraded. However, the controller remains stable and quickly responds to both the reference command and the disturbance. However, Figure 8.9 shows the PIM controller described above under the same modeling error conditions. Clearly, this controller is much more susceptible to modeling errors. While there is a region of stability and acceptable performance, instability is a real possibility and must be addressed in the design of this type of CLSS controller.

When the actual natural frequency is less than the modeled natural frequency, the OLIS/PD controller again performs well, as shown in Figure 8.10. As shown in Figure 8.11, the PIM also performs well under these conditions, remaining stable with good responses. The stability margin is larger when the modeled natural frequency (ω_m) is larger than the actual plant natural frequency (ω_a).

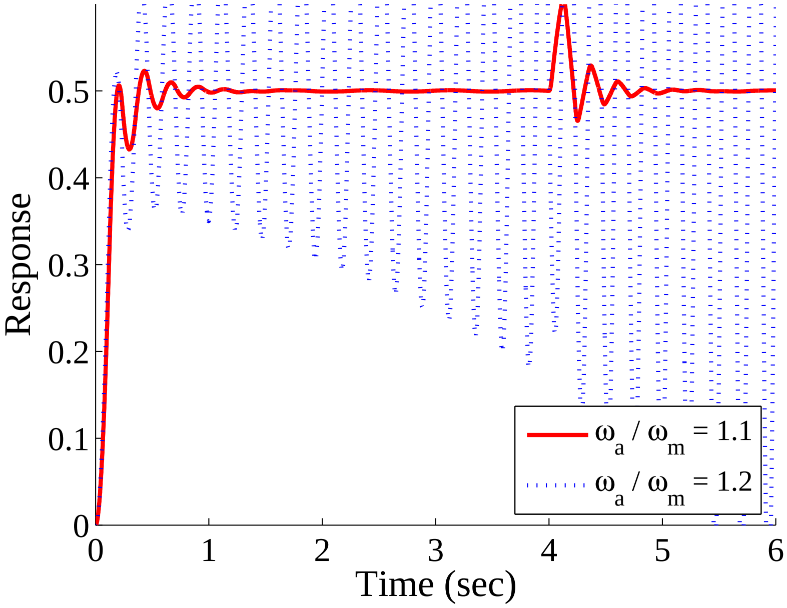


Figure 8.9: CLSS - Plant Inversion Method, $\omega_a > \omega_m$.

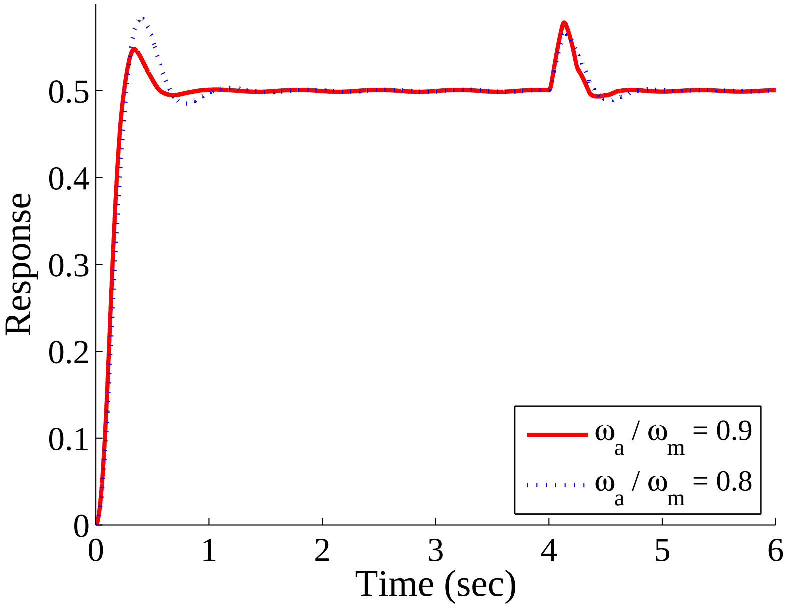


Figure 8.10: OLIS and Velocity Feedback, $\omega_a < \omega_m$.

This is actually opposite to the trend found in Chapter 6, which studied the stability of the Classical Method form of CLSS. For the Classical Method, when $\omega_a < \omega_m$, the system had small regions of stability. When $\omega_a > \omega_m$, stability margins of the Classical Method were significantly larger.

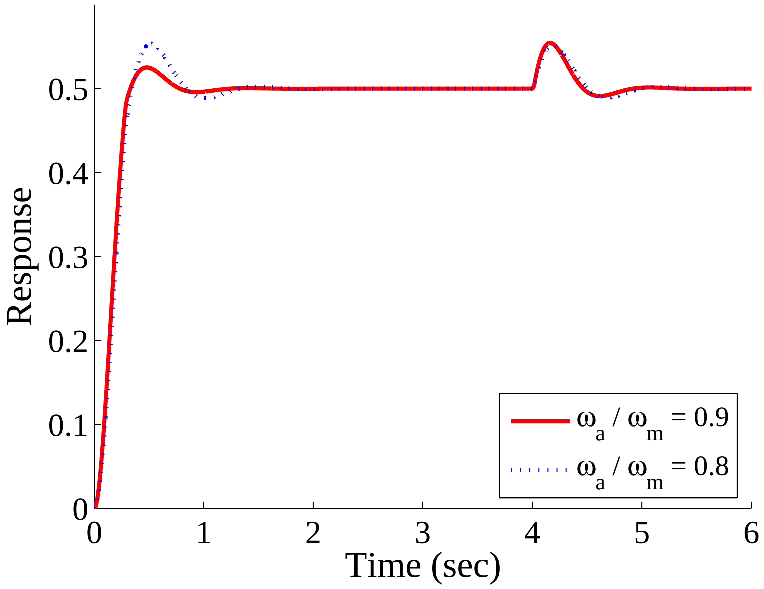


Figure 8.11: CLSS - Plant Inversion Method, $\omega_a < \omega_m$.

8.2.3 Noise Effects

While PID control has long been a useful and practical control strategy, the use of differentiators can often pose an implementation problem when real-world issues such as noise become significant. Because the PIM controller does not use any differentiators, it potentially has an implementation advantage over standard PID controllers. For example, when a small-amplitude, 60 Hz sinusoidal signal is added to the feedback signal of the OLIS/PD controller described above, the system response is virtually unchanged as shown in Figure 8.12. However, the actuator effort response shown in Figure 8.13 shows that this response is highly demanding and perhaps impossible to implement. The time scale is greatly reduced here to accurately see the oscillations. When this same feedback noise is applied to the PIM controller (using the same proportional gain used by the OLIS/PD controller), the response is as shown in Figure 8.14. Again, the response looks fairly good. However, the actuator effort required for this response is shown in Figure 8.15. The actuator demands are much higher than in the OLIS/PD case in both amplitude and frequency. Unfortunately, the PIM controller seems to be more sensitive to noise than a PD controller.

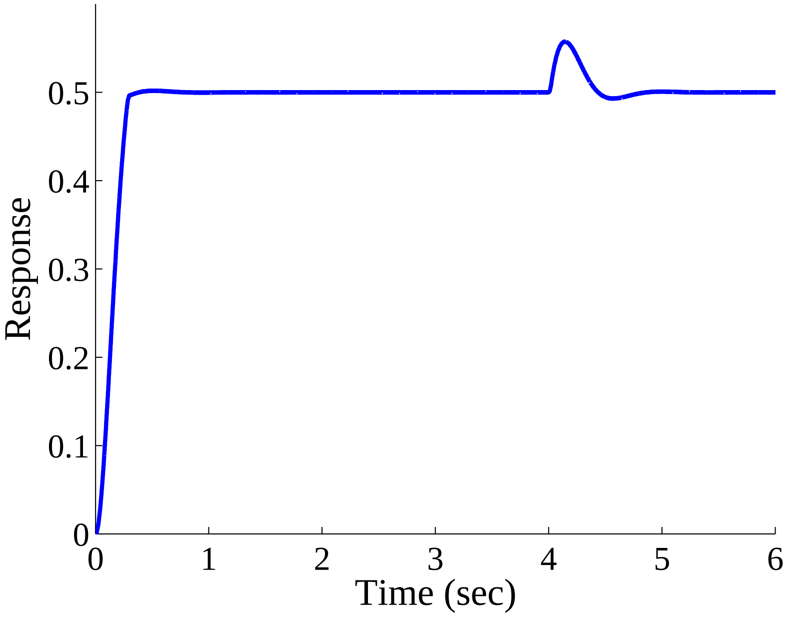


Figure 8.12: OLIS/PD System Response to Noise.

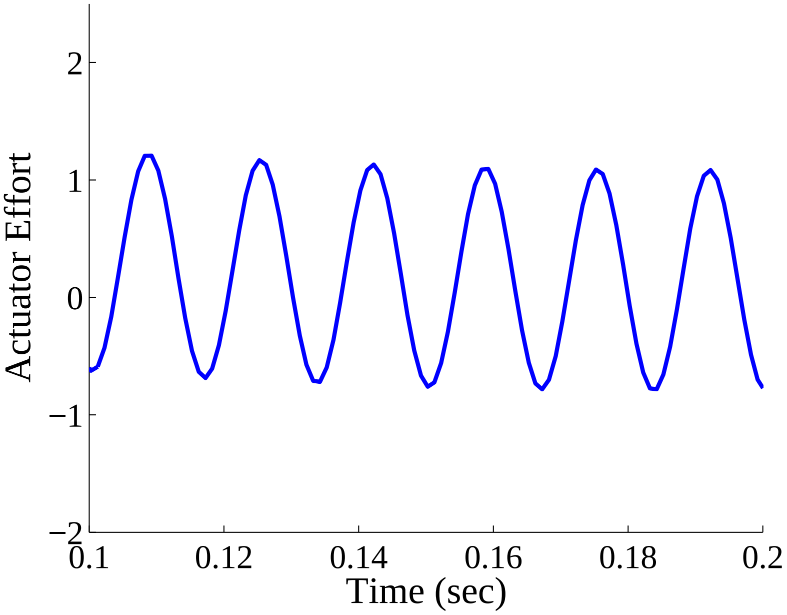


Figure 8.13: OLIS/PD Actuator Response to Noise.

Note that the actuator profile shown in Figure 8.15 is somewhat counter-intuitive, due to the near-zero output until just before $t = 0.12$ seconds. This occurs because the severe actuator response seen after $t = 0.12$ seconds is due to the portion of the PIM controller which attempts to reject disturbances via an inverted model of CG.

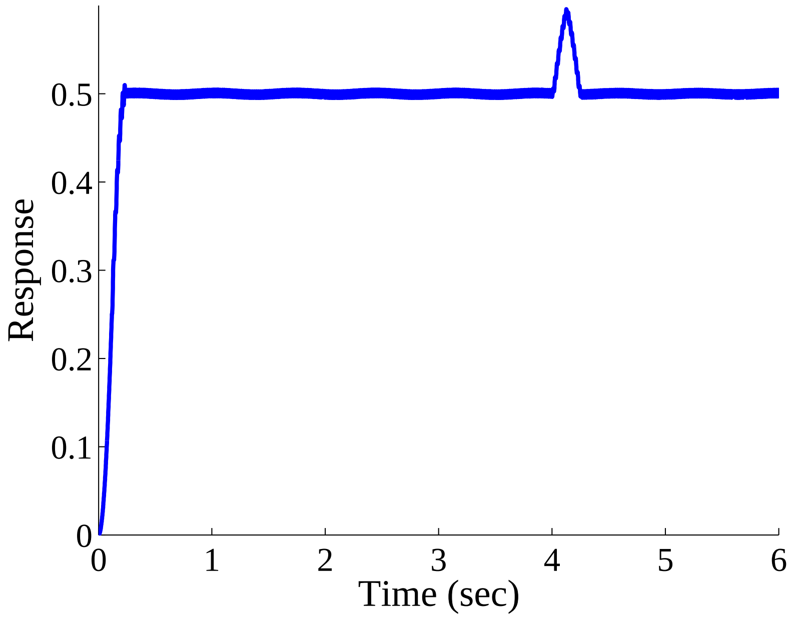


Figure 8.14: CLSS - PIM, System Response to Noise.

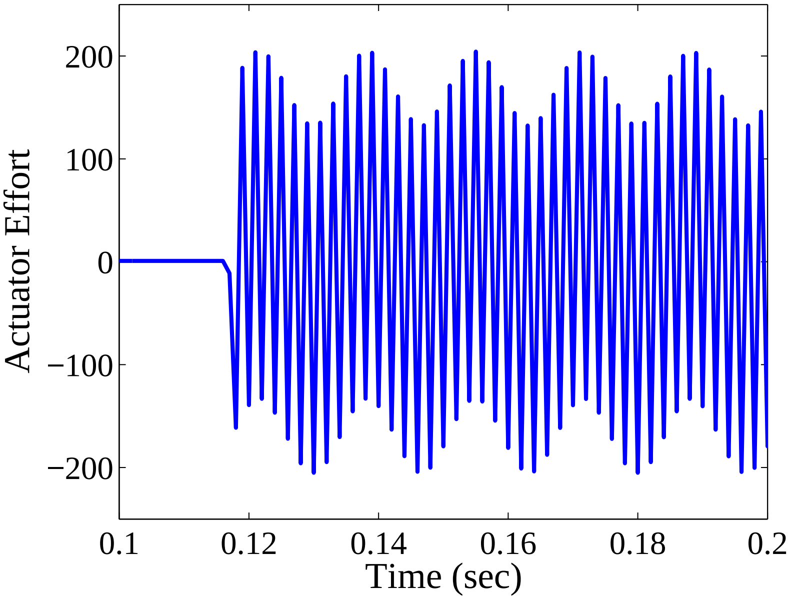


Figure 8.15: CLSS - PIM, Actuator Response to Noise.

Because the noise signal was added to the feedback line, the PIM controller treats it as a disturbance. But, the disturbance rejection portion of the PIM controller has an overall time delay, due to the I_2' term. This is why the violent reaction to the sensor noise is delayed.

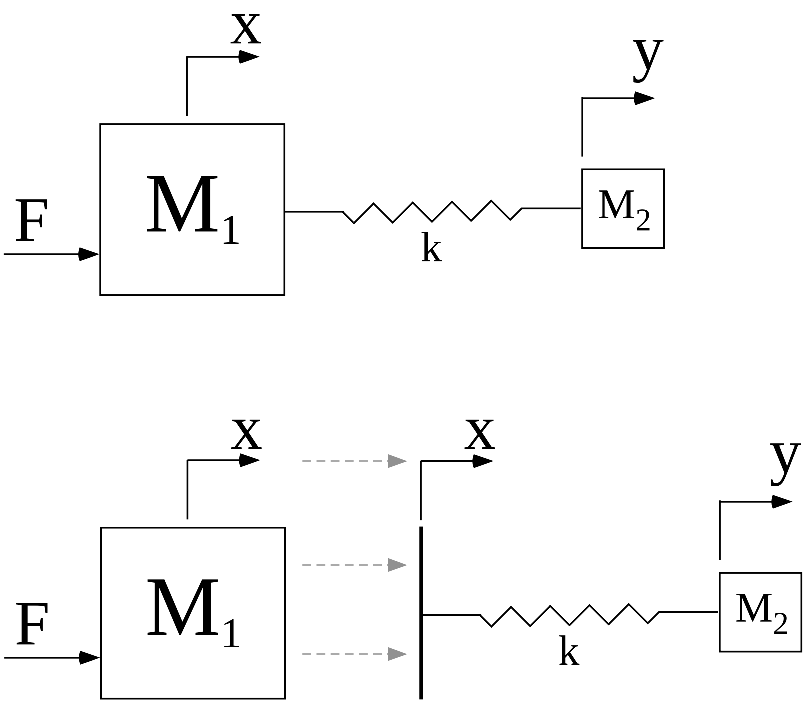


Figure 8.16: Mass-Spring-Mass System.

8.2.4 Unobservable Modes

Fortunately, there is one clear advantage to be had from using PIM over OLIS/PD controllers. That is the area of unobservable modes. In this section, the physical mass-spring-mass system shown at the top of Figure 8.16 will be studied. The actuator acts on the first mass only (M_1), and the only sensor data comes from the first mass, X. The second mass, M_2, is assumed to be small relative to the first mass. Even though they are attached via a spring, the effect of the second mass on the first is considered negligible. Therefore, the feedback controller around the first mass has a plant that is simply a scaled double integrator. The secondary transfer function which governs the motion of the second mass is a position input (position of M_1) - position output (position of M_2) transfer function. This is depicted in the lower portion of Figure 8.16, where the only force acting on M_1 is the input force F and the only action affecting the motion of the M_2 is the position of M_1.

Because there is no feedback from the second mass, its position cannot be actively

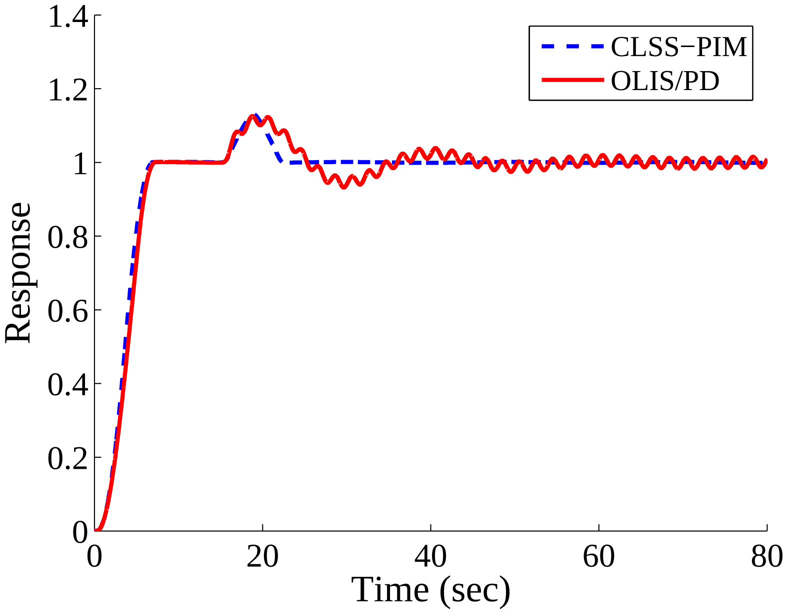


Figure 8.17: Response of Second Mass (M_2).

controlled. However, since both control schemes (PIM and OLIS/PD) filter reference commands via input shapers, the second mass can respond to reference commands without vibration. This can be seen in Figure 8.17, which shows the step reference response of the second mass (M_2) under both control strategies. This figure verifies that before any disturbances occur (before $t = 15$ seconds), both controllers (PIM and OLIS/PD control) are equally capable of providing fast, vibration free step responses.

Unfortunately, if a force disturbance acts on the first mass, this will directly affect the position of the second mass without any input shaping filter. In the absence of feedback control on the second mass, force disturbances on M_1 will usually result in uncontrolled vibration of the second mass. This can be seen in the OLIS/PD response in Figure 8.17. When a pulse disturbance acts on M_1 at time $t = 15$ seconds, the OLIS/PD controller can only eliminate one of the two system modes, the mode corresponding to the closed-loop around the first mass. The second mass vibrates continuously because the feedback controller cannot sense its motion.

However, the basic premise of PIM controllers is that responses to force disturbances are designed to eliminate vibration just as input shaping does - via destructive interference. Therefore, with knowledge of the second mode's dynamics, the force disturbance response of the PIM controller can be designed to eliminate vibration of both modes. This is evident from Figure 8.17, where the response of the second mass is vibration free even after the force disturbance.

It should be noted here that this disturbance rejection advantage only holds when D_f acts on M_1. If a disturbance acts on M_2 directly, both control schemes will be equally incapable of settling the motion of M_2. Also, by using an inverted model of the plant and controller, the use of the Plant Inversion Method with non-minimum phase plants is questionable. Even plants with zeros near the imaginary axis (in the continuous domain) or the unit circle (in the digital domain) would pose practical problems because inverting them would result in highly oscillatory poles that would cause problems in the event of modeling errors. Future work could address this problem, perhaps with an approximate plant inversion procedure similar to Zero Phase Error Tracking Control [95].

8.3 Model Reference Controller

The second, advanced CLSS controller designed for force disturbance rejection was originally developed by Ulrich Staehlin and Tarunraj Singh [91] and can be seen in Figure 8.18. This control scheme is called the Model Reference Controller (MRC) because it uses a simulated plant response (from a model of the plant, G_m) for comparison with the actual plant response to determine if a disturbance has occurred. Note that this control scheme does not require the inversion of the plant model.

One interesting thing to observe from Figure 8.18 is that the overall transfer function from the reference input to the output (assuming a perfect plant model $\rightarrow G_a = G_m = G$) is:

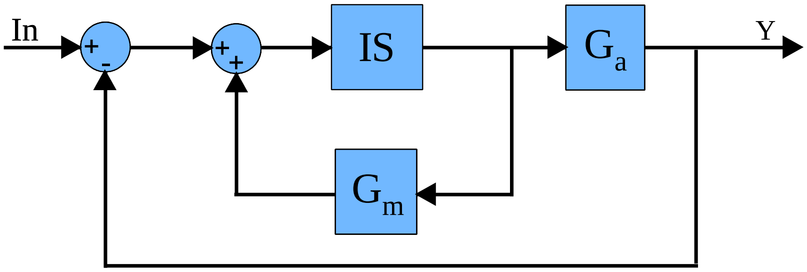


Figure 8.18: Staehlin/Singh Original Design.

$$\frac{Y}{R} = IG \tag{8.3}$$

Under perfect modeling conditions, this closed-loop system replicates standard outside-the-loop input shaping. In addition, the authors mention that one of the goals of investigating this closed-loop signal shaping controller is to obtain shaped disturbance and initial condition response. Unfortunately, their paper does not actually address this system's capabilities in these areas from either a theoretical, simulation, or experimental perspective.

8.3.1 Disturbance Rejection Capabilities - Theoretical Perspective

The ability of the MRC to reject disturbances can be established by examining the transfer function relating the Model Reference Controller's system response to a disturbance input (assuming zero modeling errors). This transfer function is:

$$\frac{Y}{D} = G(1 - IG) \tag{8.4}$$

When this equation is evaluated at the plant's poles, the transfer function becomes:

$$\frac{Y}{D} = G|_{-\zeta\omega_n \pm \omega_d j}\left(1 - \frac{0}{0}\right) = \frac{G_{num}|_{-\zeta\omega_n \pm \omega_d j}}{0}\left(1 - \frac{0}{0}\right) \tag{8.5}$$

The value of the expression within the parenthesis is undefined. There is no guarantee that this expression will equal zero, which is needed to cancel the zero from the denominator of the plant term outside the parenthesis. Unless this zero is canceled, the $\frac{Y}{D}$ transfer function will contain uncanceled poles equal to the poles of the plant, G.

The value of the $\frac{0}{0}$ term within the parenthesis can be determined by taking the limit as $s \to -\zeta\omega_n \pm \omega_d j$. If this expression goes to one, then the entire term within the parenthesis will go to zero, resulting in full plant-pole cancelation. Standard input shaping theory has never addressed this issue. Typically, as long as the IG combination evaluated at the oscillatory poles of G does not result in infinity, the actual value of $IG|_{-\zeta\omega_n \pm \omega_d j}$ is irrelevant. All that normally matters in outside-the-loop input shaping is that the poles of the oscillatory system, G, are canceled. For disturbance rejection with the Model Reference Controller, this value is now central to ensuring good performance. The following example will demonstrate that for standard input shapers, the MRC will not effectively cancel disturbances. Then, this example will show how new input shapers can be designed to not only cancel the poles of G but also ensure that $IG|_{-\zeta\omega_n \pm \omega_d j} \to 1$. This will make the MRC an effective disturbance rejecting controller.

8.3.1.1 Model Reference Control Example

Assumption:

$$G_m = G_a = G \tag{8.6}$$

Define:

$$G = \frac{\omega_n^2}{s^2 + \omega_n^2} \tag{8.7}$$

$$I = 0.5 + 0.5e^{-s\frac{\pi}{\omega_n}} \tag{8.8}$$

The input shaper shown in (8.8) is a standard ZV shaper. This shaper will cancel the poles of G; i.e. $I|_{j\omega_n} = 0$. To find the value of $IG|_{j\omega_n}$, L'Hospital's rule is used. It can be shown from this procedure that:

$$\lim_{s \to j\omega_n} IG = -\frac{\pi}{4} j \tag{8.9}$$

This means that:

$$\frac{Y}{D}|_{j\omega_n} = G|_{j\omega_n}\left(1-\frac{\pi}{4}j\right) = \frac{\omega_n^2\left(1-\frac{\pi}{4}j\right)}{0} = \infty \tag{8.10}$$

Equation 8.10 implies that a force disturbance entering a MRC controller that utilizes a standard ZV shaper will cause the closed-loop system to oscillate at the poles of the plant, G.

However, I can be re-designed to be of the form:

$$I' = Ae^{-\theta s}I = Ae^{-\theta s}\left(0.5+0.5e^{-s\frac{\pi}{\omega_n}}\right) \tag{8.11}$$

Now:

$$I'G = \frac{Ae^{-\theta s}\left(0.5+0.5e^{-s\frac{\pi}{\omega_n}}\right)\omega_n^2}{s^2+\omega_n^2} \tag{8.12}$$

L'Hospital's rule must now be used to evaluate the limit of $I'G$ as $s \rightarrow j\omega_n$. This yields:

$$\frac{-A\theta e^{-\theta s}\left(0.5+0.5e^{-s\frac{\pi}{\omega_n}}\right)\omega_n^2 - Ae^{-\theta s}\left(0.5\frac{\pi}{\omega_n}e^{-s\frac{\pi}{\omega_n}}\right)\omega_n^2}{2s} \tag{8.13}$$

Evaluated at $s = j\omega_n$, this fraction becomes:

$$\frac{-0.5\pi A\left(e^{-\theta\omega_n j-\pi j}\right)}{2j} \tag{8.14}$$

If $A = \frac{-4}{\pi}$ and $\theta = \frac{\pi}{2\omega_n}$, then this fraction will become unity. This means that the $\frac{Y}{D}$ transfer function (as $s \rightarrow j\omega_n$) becomes:

$$\frac{Y}{D}|_{j\omega_n} = G|_{j\omega_n}(1-1) = \frac{0}{0} \neq \infty \tag{8.15}$$

This $\frac{0}{0}$ term is not equal to infinity because G is only one mode, and therefore it does not have multiple poles at any given $s = \sigma + j\omega$ location on the real-imaginary plane.

This example showed how MRC controllers utilizing standard input shapers will generally fail to reject force disturbances. However, this example also showed how new input shapers can be designed to allow a MRC controller to effectively reject disturbances. The example shown here developed the new input shaper shown in (8.11).

This new input shaper was a ZV shaper scaled by A and delayed by θ seconds. While this new input shaper did ensure that the $\frac{Y}{D}$ transfer function would not oscillate at the poles of the plant (G), it also eliminated one of the key characteristics of input shapers: the amplitudes no longer add to unity. In fact, in the example shown, the amplitudes add to A. It is important that the impulses comprising an input shaper add to unity, so that the system will reach the same output that was originally desired by the unshaped command.

To resolve this dilemma, the derivation of input shapers can easily be modified. The derivation of input shapers normally includes a constraint that the impulse amplitudes sum to unity; along with other constraints concerning vibration suppression, robustness, etc. For input shapers used in a MRC controller, the shaper design procedure simply needs to be augmented to include the additional constraint that $IG|_{-\zeta\omega_n \pm \omega_d j} \rightarrow 1$. This will, most likely, require additional impulses and a larger shaper duration. The examples given in the subsequent sections use newly designed input shapers within MRC controllers whose impulses sum to unity.

8.3.2 Model Reference Controller Performance - Simulations

When this new input shaper, I', is used in place of the standard input shaper, I, the control scheme shown in Figure 8.18 will eliminate residual vibration even when the system experiences reference inputs, force disturbances, sensor disturbances, and/or non-zero initial conditions. Outside-the-loop input shaping only eliminates residual vibration caused by reference inputs.

In order to demonstrate the effectiveness of this control scheme, a set of simulations was conducted. These simulations compared the original MRC controller using standard input shapers, the modified MRC controller using the new input shapers designed in this dissertation, and a current, standard controller. The standard controller chosen for comparison, which can be seen in Figure 8.19, utilizes an input

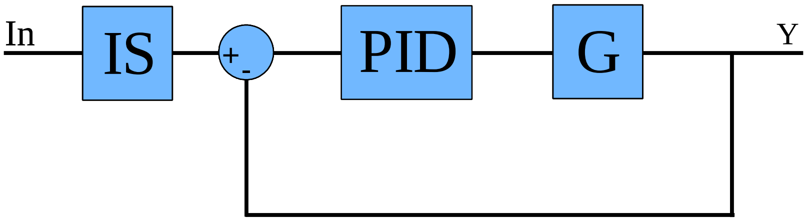


Figure 8.19: OLIS/PID Controlled System.

shaper outside of the feedback loop and a PID controller inside the loop. While some time was spent tuning the PID controller, the PID gains were not rigorously optimized relative to a set of performance specifications. Therefore, it may be possible to create an OLIS/PID controller that outperforms the one shown here.

In these simulations, four basic responses are studied: a step reference input, non-zero position and velocity initial conditions, a pulse force disturbance, and a pulse sensor disturbance. Recall that a force disturbance enters the block diagram just before the actual plant, while a sensor disturbance enters the block diagram at the feedback signal. In addition, the effect of modeling errors in natural frequency was studied. Finally, it should be noted that only undamped, second-order systems were analyzed here.

The first set of responses, shown in Figure 8.20, show step responses with zero initial conditions, a force disturbance at $t = 5sec$, and a sensor disturbance at $t = 10sec$. The fastest step response is realized by the original Model Reference Controller which utilizes standard input shapers. However, this method fails to reject either type of disturbance. On the other hand, the tuned PID controller quickly rejects both disturbances. Unfortunately, it has a significantly longer settling time to reference inputs as compared to either of the Model Reference Controllers. The modified Model Reference Controller (which utilizes the newly designed input shapers) produces the best overall performance. It has a reference input settling time less than half that of the outside-the-loop input shaping/PID controller and much better disturbance rejection than the original Model Reference Controller.

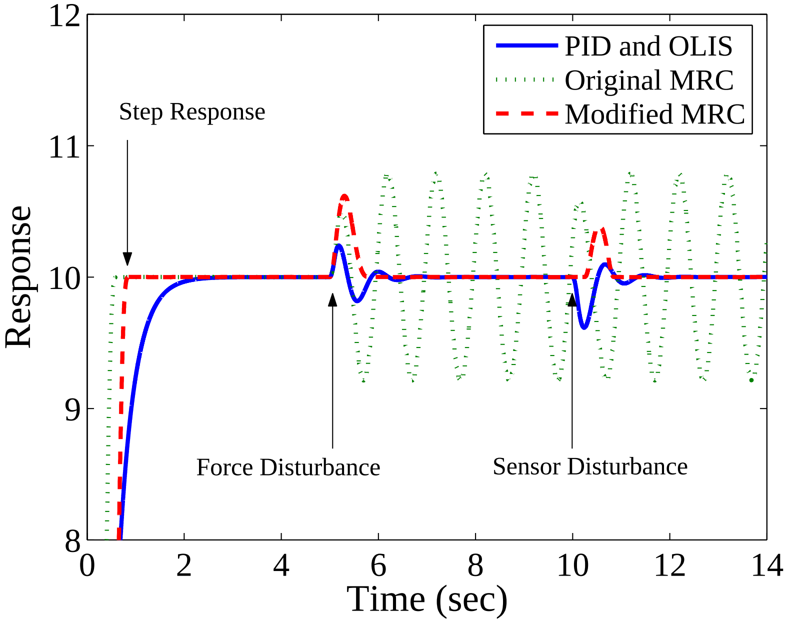


Figure 8.20: Step Response with Zero Initial Conditions.

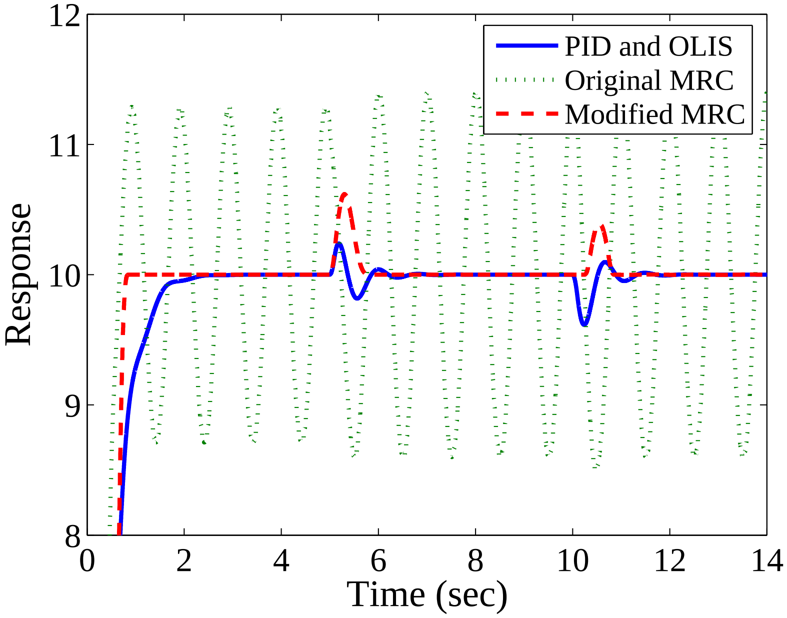


Figure 8.21: Step Response with Non-Zero Initial Conditions.

Adding non-zero position and velocity initial conditions reinforces these findings. Figure 8.21 shows the same controllers and inputs as in the previous figure. However, the actual plant, G_a, was given an initial position of 1 unit and an initial velocity of 1 unit per second. The original Model Reference Controller utilizing a standard input

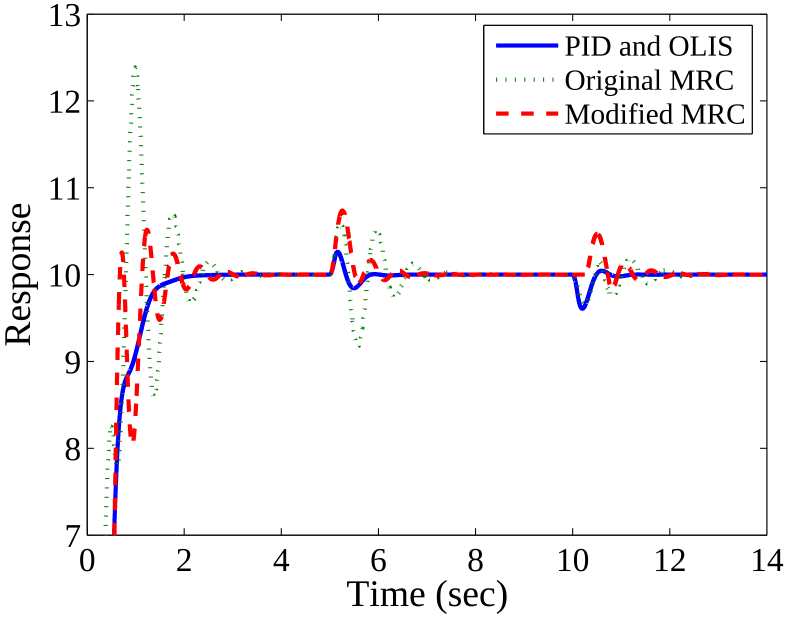


Figure 8.22: Step Response with $\frac{\omega_a}{\omega_m} = 1.2$.

shaper fails to mitigate the oscillations induced by the initial conditions. In addition, the step response of the outside-the-loop input shaping/PID controller continues to require a relatively long settling time. On the other hand, when the new input shaper design is utilized within the Model Reference Controller, an excellent step response is achieved despite the non-zero initial conditions.

The last set of simulations was intended to observe the effect of natural frequency modeling errors. When the actual, natural frequency is 20% higher than the modeled natural frequency ($\frac{\omega_a}{\omega_m} = 1.2$), the outside-the-loop input shaping/PID controller performs best. This can be seen in Figure 8.22. This result was anticipated, due to the fact that a PID feedback controller relies much less on a system model than either of the MRC controllers.

As can be seen in Figure 8.23, when the actual, natural frequency is 20% lower than the modeled, natural frequency ($\frac{\omega_a}{\omega_m} = 0.8$), the original Model Reference Controller utilizing a standard input shaper goes unstable. This was predicted by Staehlin and Singh [91]. But, as shown in Figure 8.24, utilizing the newly designed input shaper

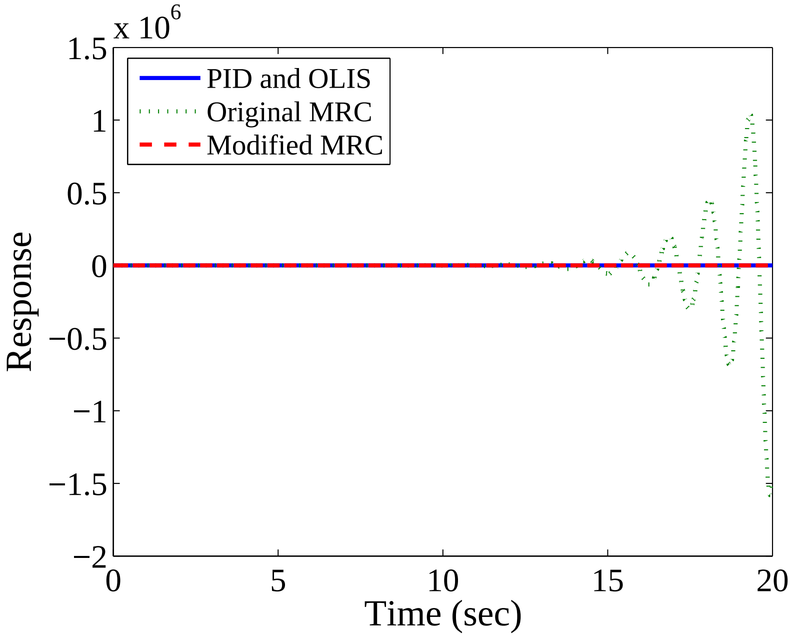


Figure 8.23: Step Response with $\frac{\omega_a}{\omega_m} = 0.8$.

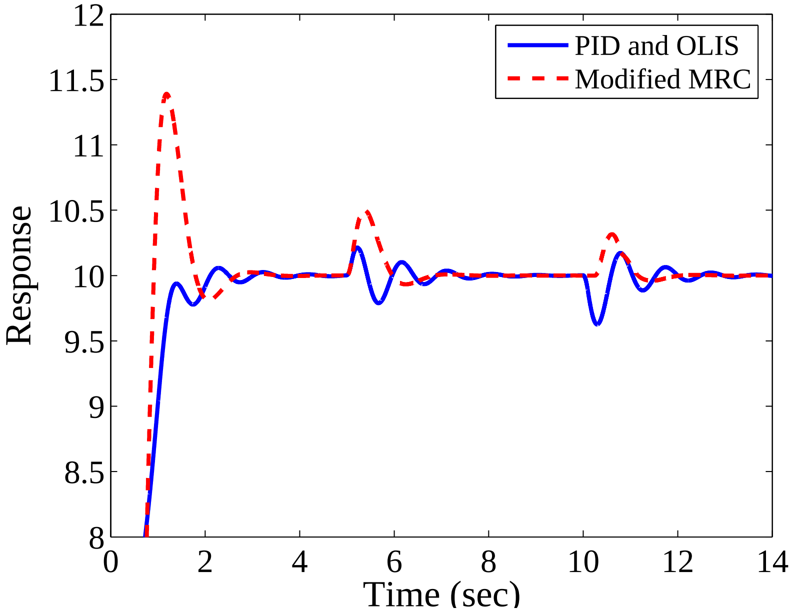


Figure 8.24: Modeling Error $\left(\frac{\omega_a}{\omega_m} = 0.8\right)$ with the Original MRC Data Removed.

within the Model Reference Controller makes the controller stable. Secondly, under this modeling error condition, the outside-the-loop input shaping/PID controller does not clearly outperform the modified Model Reference Method. While the outside-the-loop input shaping/PID response reaches the desired set point without overshoot, its settling time is comparable to the modified Model Reference Controller. In addition, the modified Model Reference Controller rejects disturbances quicker than the outside-the-loop input shaping/PID control scheme. Although, the overshoot due to the disturbance is higher for the MRC controller. In summary, neither the OLIS/PID controller nor the MRC controller utilizing the newly designed input shaper has a clear performance advantage when there are natural frequency modeling errors.

Finally, a note should be made here concerning some practical advantages of using MRC control. The examples given in this section show that the MRC controller is capable of quick step responses (even when non-zero initial conditions are present) and disturbance rejection. And, all of this is accomplished without the need for differentiators or integrators. The actual implementation of differentiators and integrators can be challenging, especially when sensor noise and integrator windup are considered. The MRC controller, therefore, presents a useful alternative to PID feedback control. However, the MRC controller is more dependant upon an accurate system model. Therefore, when accurate system models are not available, a standard PID controller will, most likely, be the best choice.

8.4 Model Reference Controller Applied to a Spacecraft Model

This section will describe the application of the Model Reference Controller scheme to a spacecraft model provided by an aerospace industry research partner. The spacecraft is modeled in MATLAB as a 25-mode system with 12 inputs and 42 outputs. The twelve inputs are realized by two actuators located at distinct points on the spacecraft. Each actuator is able to produce a linear force and a moment in the three cartesian directions. The 42 outputs are actually the six position measurements (linear distance and rotation) of seven distinct points on the spacecraft.

Initial investigation into the dynamics of this spacecraft revealed that one of its 25 modes was dominate. Six of these modes were rigid-body modes and that the dominate mode discussed here was the dominate oscillatory mode.

As a demonstration of the Model Reference Controller and its ability to reject disturbances, a simple maneuver was studied. This maneuver used the z-direction, linear force of the actuator located close to the spacecraft's geometric center. The force was used to maneuver the tip of one of the spacecraft's flexible appendages from the 0 meter position to the 1 meter position. Each of this section's remaining, simulated responses shows the z-direction motion of this appendage's tip.

8.4.1 Outside-the-Loop Input Shaping

The block diagram shown in Figure 8.25 was used to demonstrate outside-the-loop input shaping. The block labeled "Signal Generator" created a step reference input of 1 meter. The block labeled "Input Vector" created a 12 X 1 force input vector. The "Rigid Body" block created a bang-coast-bang input profile to create a point-to-point motion. Finally, the "Input Shaper" block filtered the input command to eliminate the dominant vibratory mode of $0.25Hz$. The disturbance force is the signal denoted by D_f.

Without the disturbance force disrupting the motion of the spacecraft, the z motion of the appendage's tip makes an excellent response. This can be seen in

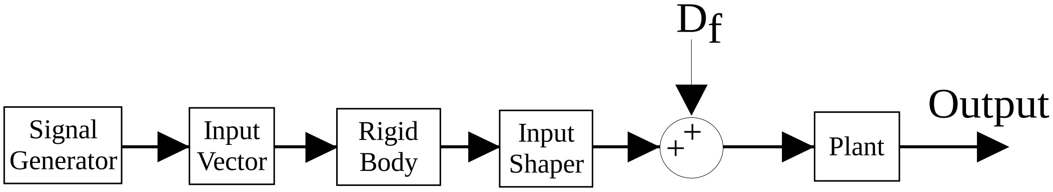


Figure 8.25: Outside-The-Loop Input Shaping Block Diagram.

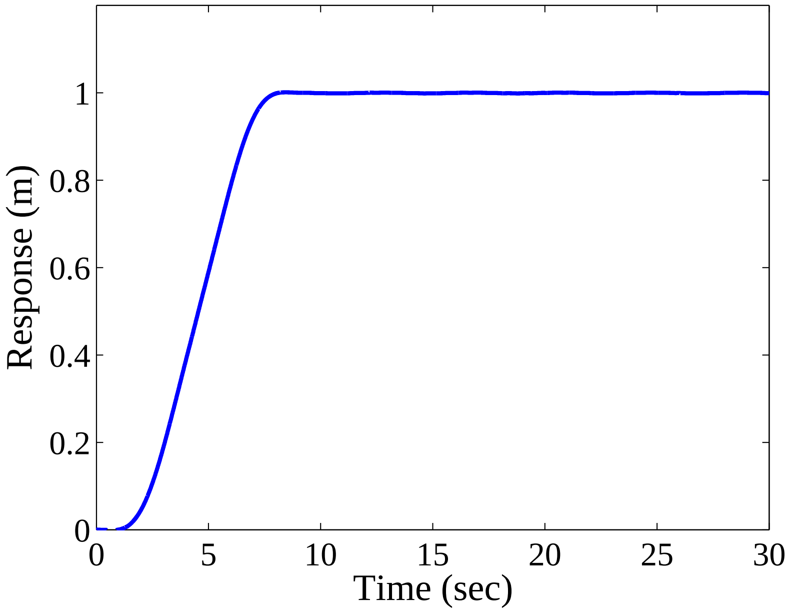


Figure 8.26: Outside-The-Loop Input Shaping Without Disturbance.

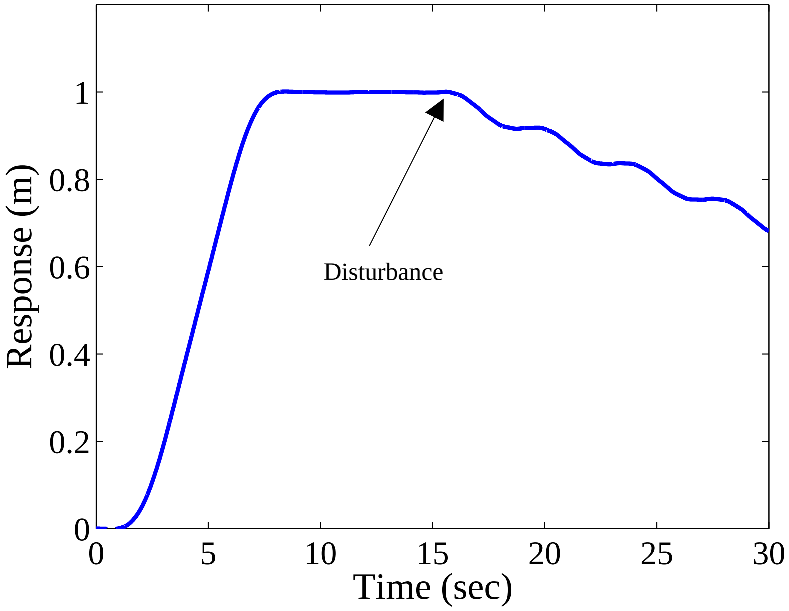


Figure 8.27: Outside-The-Loop Input Shaping With Disturbance.

Figure 8.26. However, when a quick (0.2 seconds duration) pulse disturbance force is added to the block diagram at time $t = 15sec$, this controller is unable to maintain a constant, non-vibratory position. The response, as seen in Figure 8.27, contains a non-zero velocity and oscillates at the dominant frequency.

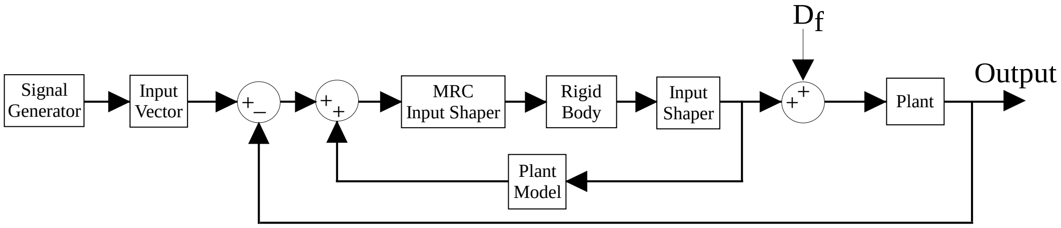


Figure 8.28: Model Reference Control Block Diagram.

8.4.2 Closed-Loop Signal Shaping Control - Model Reference Controller

The control scheme utilized to achieve disturbance rejection is the MRC controller shown in Figure 8.28. Within this block diagram, the “Signal Generator”, “Input Vector”, “Rigid Body”, and “Input Shaper” blocks serve the same purposes as in Figure 8.25. The new addition is the block labeled “MRC Input Shaper”, which is necessary to achieve a vibration free response to a disturbance. In Section 8.3.1.1, the blocks corresponding to “MRC Input Shaper” and “Input Shaper” were designed together, as one block. They are shown separately in these figures only to emphasize that the block “MRC Input Shaper” is the advancement made beyond Staehlin and Singh’s original design.

Using the same reference input and the same disturbance force (this time, the disturbance was applied at time $t = 12sec$), the response of the control system sketched in Figure 8.28 is shown in Figure 8.29. There are a few important things to notice from this response. First, the disturbance does not result in significant vibration or steady drift away from the desired location. However, the position of the node does settle to a position slightly offset from the desired 1 meter location. And, unfortunately, there is also a large cost in overshoot. In fact, the overshoot for this example is approximately 150%.

Fortunately, this overshoot problem can be addressed. In addition to rejecting disturbances, this control scheme was also shown to be capable of eliminating the

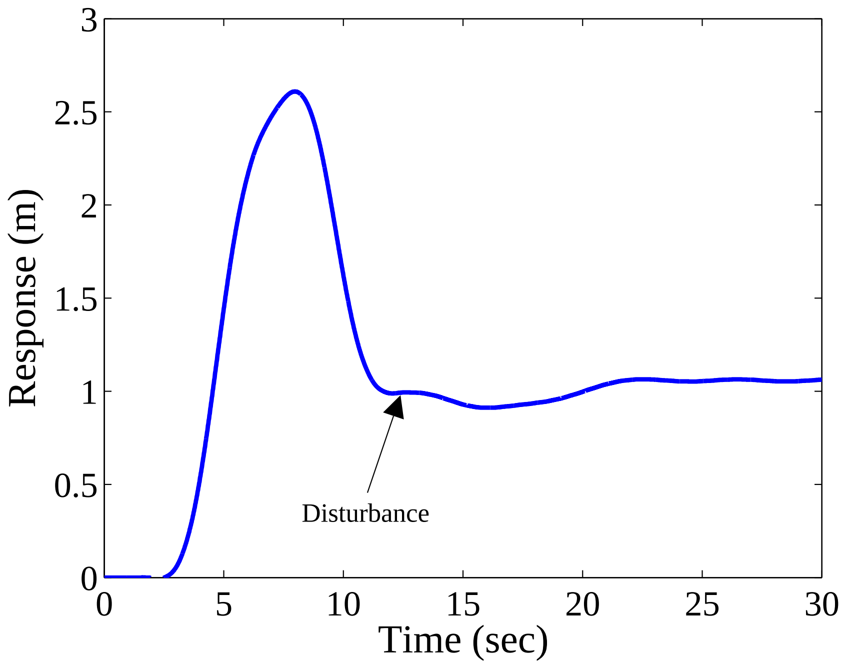


Figure 8.29: Model Reference Controller Response.

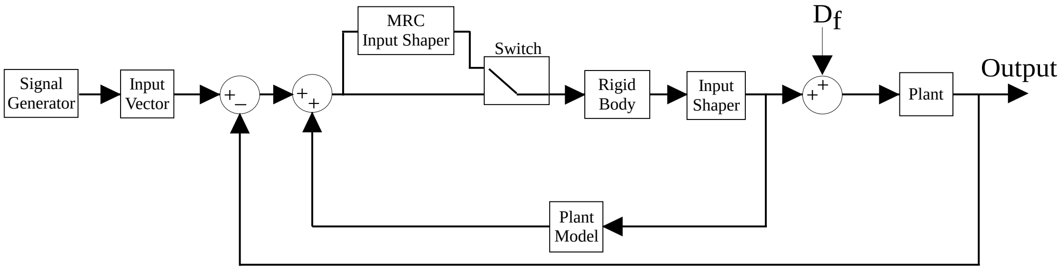


Figure 8.30: MRC Block Diagram with Switching Mechanism.

effects of non-zero initial conditions. This was shown in Section 8.3.2. This knowledge enables a slight modification of the MRC controller, producing the new MRC controller shown in Figure 8.30. This control scheme assumes that the time required to complete the 1 meter move is approximately known. Given a model of the spacecraft and actuator properties, this should not be an unrealistic assumption. During the intended motion, the "MRC Input Shaper" control block is disconnected. It is then reconnected after the intended motion is complete. If there are no disturbances during the intended motion, then the system will respond exactly as it did in the open-loop response shown in Figure 8.26. If a disturbance enters the system after

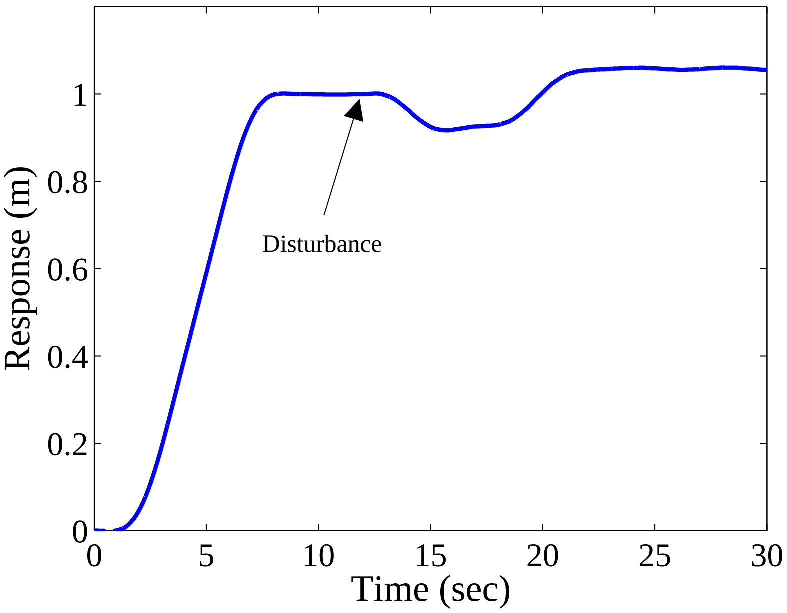


Figure 8.31: Response of Control Scheme with Switching Device.

the intended motion is complete and after the "MRC Input Shaper" block has been switched on, then the system will have already reached its desired position without overshoot and will then simply reject the disturbance. An example of this can be seen in Figure 8.31. Again, the disturbance is applied at time $t = 12sec$. Even though this control scheme has an excellent step response and successfully rejects the disturbance in terms of residual vibration, there is still a steady-state error problem.

However, should the disturbance occur before the intended motion is complete and/or before the "MRC Input Shaper" block is switched on, then the controller will simply view the system's non-stationary, vibratory response as non-zero initial conditions. Once the "MRC Input Shaper" block is eventually turned on, the system will be brought to rest. This can be seen from Figure 8.32. Here, the disturbance occurs at time $t = 2sec$, well before the intended motion could have been completed.

The two previous responses show how the MRC controller can reject disturbances. However, the system eventually settles at a position not equal to the reference position. Unfortunately, this is a characteristic inherent to MRC controllers that have

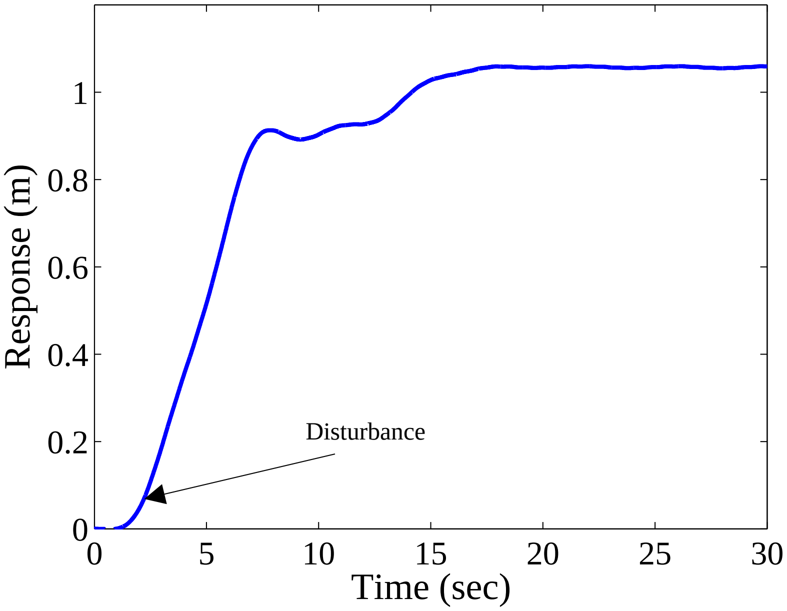


Figure 8.32: Second Response of Control Scheme with Switching Device.

plants with rigid body modes. Fortunately, the MRC controller is able to greatly diminish the drift that would normally occur due to a force disturbance (as shown in Figure 8.27).

This position offset caused by the disturbance can be dealt with by altering the reference input once the offset is detected. For example, Figure 8.33 shows the same response given in Figure 8.32, except that the reference input is altered at time $t = 30sec$ to reset the spacecraft to its desired position.

As was noted in Section 8.3.2 and in [91], MRC controllers can be sensitive to modeling errors. The spacecraft model used in this section is a 12 input, 42 output system given in state-space form. The plant model used to develop the MRC controller approximated the spacecraft as a mass-spring-mass type system: a plant with a rigid body mode and one vibratory mode. This was justified from simulation responses that showed a clear, dominant mode of vibration. In addition, the design procedure for MRC controllers (which was developed in Section 8.3.1.1, depended upon an accurate transfer function of the plant. The state-space system describing the spacecraft was

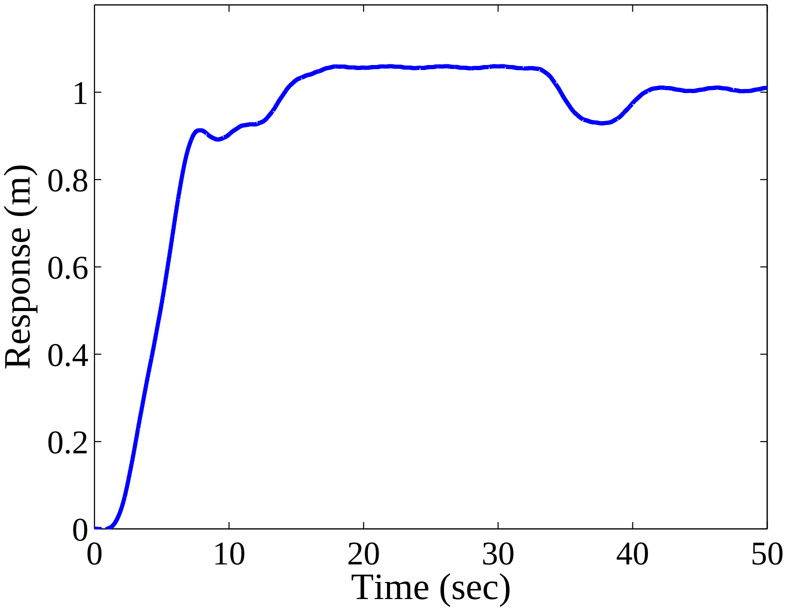


Figure 8.33: Altering Reference Input to Eliminate Offset.

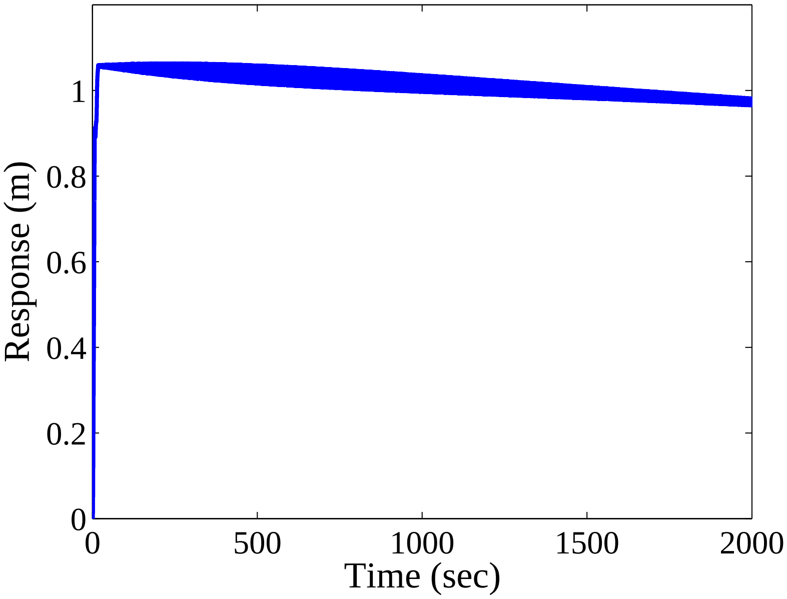


Figure 8.34: Beating Frequency Vibrations due to Modeling Errors.

too large and complex to convert to the laplace domain. Therefore, some of the plant parameters were estimated from the full, spacecraft model's responses.

Using a simplified model and estimating parameters created modeling error, which resulted in incomplete vibration cancelation. For example, Figure 8.34 shows the response given in 8.32 carried out to 2000sec. This figure shows incomplete vibration suppression and a slight non-zero drift. The vibration shown here actually forms a beating response, not exponentially growing oscillations.

There are several ways in which this vibration and drift problem can be addressed. The first method is to develop a more accurate model of the plant. The second is to extend the development of MRC controllers to include robust input shapers. The third approach is to create an algorithm that intelligently turns the MRC controller on and off at appropriate times. This approach will be chosen because it is simpler than theoretically advancing the MRC derivation to include robust shapers. In addition, this approach accepts the presence of modeling errors, instead of trying to eliminate them with increasingly accurate models which are not always practical to obtain. This algorithm is similar to the switching mechanism that was depicted in Figure 8.30, but also includes the ability alter the reference command, as was shown in the response depicted in Figure 8.33.

Figure 8.34 shows that after the disturbance is rejected, the vibrations caused by the modeling errors are small. They then grow over time, before shrinking again. If the intelligent algorithm controlling the MRC controller turns the actuator inputs off once a disturbance has been successfully rejected, these vibrations will remain small. The result can be seen in Figure 8.35, where a disturbance was first rejected and the actuator effort was then turned off. This figure shows that the vibration is much smaller than shown in Figure 8.34. However, a small drift is still present. The drift created by modeling errors, as well as offsets created by the natural disturbance rejection, can be eliminated by turning the actuator effort back on and intelligently altering the reference input. This was shown in Figure 8.33.

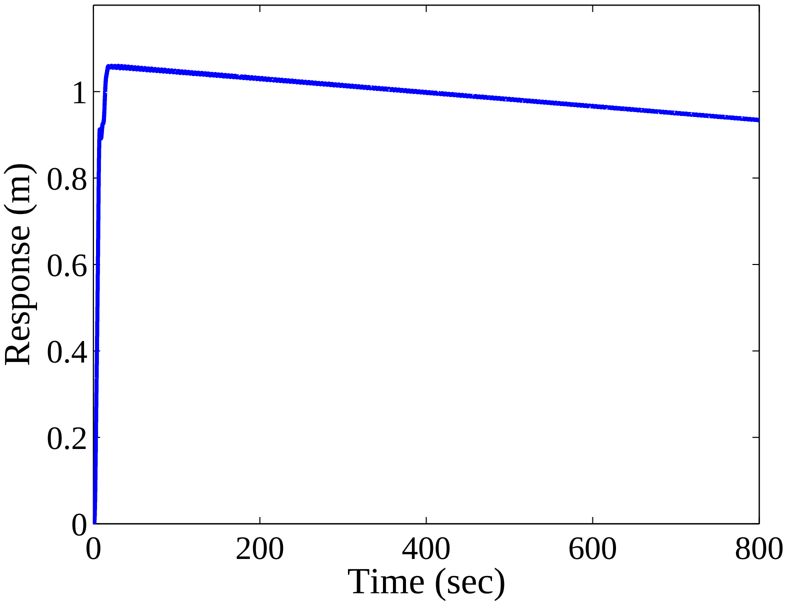


Figure 8.35: Switching Off the Actuator Inputs On the MRC Controller.

8.4.3 Discussion of Advanced CLSS Controllers

This chapter advanced the use of CLSS controllers beyond the simple, Classical Method form. The purpose of this advancement was to find CLSS controllers that are capable of rejecting force disturbances. The first approach discussed was the Plant Inversion Method (PIM). This controller was shown to be theoretically capable of rejecting force disturbances, even when an unobservable, second mode existed. This is a clear advantage over PID control. However, the use of PIM controllers is questionable, due to the plant inversion requirement. This leads to noise and model sensitivity.

The study of Model Reference Controllers (MRC) also showed that placing input shapers inside of feedback loops can be used to reject force disturbances and the effects of non-zero initial conditions. However, as was discussed in Section 8.3.2 and Section 8.4, MRC controllers also rely upon accurate system models. It should be noted that some of the modeling error problems associated with MRC control were addressed via an intelligent switching algorithm.

The advantages and disadvantages of Model Reference Controllers and Plant Inversion Method controllers establish a trade-off between these and PID controllers. PID controllers do not rely heavily on accurate system models. However, they do require the implementation of differentiators and integrators. In certain applications, this can be a significant challenge. However, applications where significant modeling errors are expected will probably necessitate the use of PID control. The practical use of PIM and MRC controllers should be reserved to special situations where accurate models are obtainable and PID control is not practical to implement or does not produce adequate performance.

CHAPTER IX

CONCLUSIONS AND FUTURE WORK

9.1 Conclusions and Thesis Contributions

9.1.1 Concurrent Design of OLIS and PID Feedback Controllers

The first part of this dissertation concentrated on the concurrent design of outside-the-loop input shapers and PID feedback controllers. The major contribution offered by this research was to definitively show that concurrently designing OLIS/PID controllers yields unique and superior controllers when the design constraints do not severely restrict the controller (as when the design constraints force an over-damped system). And, this superiority is not restricted to any particular controller design strategy. This thesis examined analytical based solutions, numerical search routines, general rule-of-thumb techniques, and numerical optimizations.

This contribution can be divided into several smaller contributions. The first result offered by this research is a detailed understanding of how and why concurrently designed OLIS/PID feedback controllers are superior to sequentially designed combinations. The major reason is that input shaping reduces vibrations that are induced by the reference command. Because the input shaping filter outside of the loop can reduce reference-induced vibration, the feedback controller is relieved of this responsibility, and is allowed to prioritize other performance indices (like rise time). The result is that the feedback controller can be more aggressive than would be allowable if only feedback control were used, resulting in an overall controller (OLIS and PID feedback control) with a faster response.

The second result in this area was the development of fast and simple search algorithms for concurrently designing OLIS/PID controllers for simple plant types.

These results were verified via simulation and experiment. Finally, these algorithms were shown to be effective on multi-mode plants that can be approximated as single-mode systems.

The third result offered by this thesis in the area of concurrent design of OLIS/PID controllers was a two-pronged approach to concurrent design when the plant is a complicated system that cannot be approximated by a simple model. The first design approach used newly designed PID tuning rules that yield aggressive closed-loop systems. When combined with input shaping, the overall controllers are superior to those designed by the Ziegler-Nichols tuning rules. For more precise concurrent designs, a numeric optimization routine was developed. This optimization routine solved for as many parameters (and verified as many constraints) as possible via a linear optimization routine. The remaining parameters and constraints were designed and met via a nonlinear optimization routine. The overall optimization yielded more-consistent and superior controllers than did a standard nonlinear optimization package that attempted to solve for all parameters (input shaper and PID) and meet all constraints.

9.1.2 Closed-Loop Signal Shaping Controllers

The second half of this dissertation investigated feedback controllers that contained an input shaping filter somewhere within a feedback loop. The major contribution was to show that CLSS controllers do have some useful applications, but that their need for accurate system models limits their practical usage. The majority of the previous literature focused on applications where CLSS controllers are useful. This dissertation specifically details several applications and scenarios when CLSS is not a desirable control strategy. And, this dissertation expands the discussion of good, practical uses of CLSS controllers.

This major contribution is divided into the following areas. The first contribution this dissertation provided in the area of CLSS controllers was a detailed stability analysis based upon classical control theory. As mentioned in Chapter 1, the literature contains several CLSS stability studies. However, they are primarily reliant upon more complicated stability theorems, like the Lyapunov stability criterion. While these studies are certainly useful, this dissertation investigated the stability of CLSS controllers via the root locus and Bode plots. These classical control tools are fairly easy to understand and often give more insight, or intuitive feel, to the system being studied than is given by some of the more complicated stability testing methods.

In addition to showing basic root loci and Bode plots for CLSS controllers, this dissertation studied the effect of modeling errors, system damping, and the addition of lead compensators. Finally, the stability study verified some of its findings experimentally.

The major result from the stability analysis is that Classical Method CLSS controllers can be stabilized via simple proportional or lead compensator control. However, the sensitivity to modeling errors and the extra dynamics added by the input shaper make the practical choice of CLSS controllers over standard PID control somewhat limited. Two important exceptions to this are the successful use of a CLSS controller on the Georgia Tech HighBay crane and the use of human-centered, CLSS controllers on cranes.

The second contribution in the area of CLSS was to establish a variety of useful applications for these types of controllers. This thesis showed how Classical Method CLSS controllers can be used to reject sensor disturbances, to stabilize and improve the performance of non-collocated controllers, to handle actuator saturation, and to improve the performance of human-operated machinery. This thesis also verified the improved performance of human-operated machinery via an experimental study.

The third contribution was the development of advanced CLSS controllers. This thesis investigated two advanced CLSS controllers from the previous literature: the

Plant Inversion Method (PIM) and the Model Reference Controller (MRC). The PIM was first improved by solving the inherent causality problem. Secondly, this form of CLSS was shown to be effective at eliminating the effect of disturbances on systems with an unobserved mode. The MRC was first improved by designing new input shapers that actually make it possible for the MRC to reject force disturbances. This new CLSS controller was then tested on a high-fidelity simulation of a spacecraft system.

Both of these advanced, CLSS controllers show potential advantage over OLIS/PID control. Neither CLSS controller requires the implementation of differentiators, and the PIM controller can eliminate the effect of force disturbances on unobservable modes. However, their sensitivity to noise and modeling errors makes their practical implementation questionable. Future research will need to be done to determine if these problems can be successfully mitigated.

9.2 Future Work

In the field of concurrently designing outside-the-loop input shaping and PID feedback control, the application to generally complex plants is one, major area where improvements can be made. Section 5.2.2.6 highlighted most of the current deficiencies in the solution provided by this dissertation. Included among these issues was computation time and failure to find a solution. While the procedure developed in this dissertation is a good tool, the problems reviewed here provide ample research questions for future work. Finally, while this dissertation showed how the provided solution can be superior to a generic, nonlinear optimization package, a rigorous proof of this superiority would be an excellent contribution.

A second opening for future work in this field is the study of approximate, second-order systems. Future work could study other plant types (including those with three or four oscillatory modes, first-order dynamics, and non-minimum phase plants).

Also, future work could specifically address the effects of a numerator zero becoming non-negligible. These studies would add to the general understanding of when the second-order approximation is useful.

The most promising area of future research on closed-loop signal-shaping (CLSS) controllers lies in the field of force disturbance rejection. Chapter 8 showed how the PIM and MRC forms of CLSS will reject force disturbances. However, PIM's use of plant inversion makes its practical use questionable. For example, this dissertation showed that this control scheme is sensitive to noise and modeling errors, as well as being currently impractical for use with non-minimum phase plants. The MRC form of CLSS did not require plant inversion. However, when rejecting a force disturbance, this controller had a natural steady-state error problem. This issue was partially addressed within this dissertation. However, a focused study of this issue could yield an improved MRC control scheme that does not suffer from this steady-state error problem.

A second promising area for future research on CLSS controllers is in the area of stability analysis. It will be advantageous to continue the stability study with more complicated plants: including non-minimum phase plants.

Apart from improving the CLSS controllers studied in this dissertation, there is the need to develop and study new forms of CLSS. To date, only four significantly different CLSS controllers have been seriously addressed either in the literature or in this dissertation. While these previously studied CLSS controllers have been shown to be useful in various applications, there will always be a need to develop new forms of CLSS controllers that have their own unique advantages.

REFERENCES

[1] Agostini, M. J., Parker, G. G., Groom, K., Schaub, H., and Robinett, R. D., "Command shaping and closed-loop control interactions for a ship crane," in *American Controls Conference*, 2002.

[2] Alli, H. and Singh, T., "On the Feedback Control of the Wave Equation," *Journal of Sound and Vibration*, vol. 234, no. 4, pp. 625–640, 2000.

[3] Banerjee, A., Pedreiro, N., and Gonzalez, M., "Simultaneous Optimization of Input Shaping and Feedback Control for Slewing Flexible Spacecraft," in *American Control Conference*, (Denver, CO), pp. 4796–4798, 2003.

[4] Calvert, J., Wilmette, and Gimpel, D., "Method and Apparatus for Control of System Output in Response to System Input," *U.S. Patent No. 2,801,351*, 1957.

[5] Calvert, J. F. and Sze, T. W., "Short-Time Memory Devices in Closed-Loop Systems," *U.S. Patent No. 3,010,035*, 1961.

[6] Chang, P. H. and Park, J., "Use of Input Shaping Technique with a Robust Feedback Control and Its Application to the Position Control of Surface Mount Machine," in *IEEE International Conference on Control Applications*, (Dearborn, MI), pp. 397–402, 1996.

[7] Chang, P. H. and Park, J., "A concurrent design of input shaping technique and a robust control for high-speed/high-precision control of a chip mounter," *Control Engineering Practice*, vol. 9, pp. 1279–1285, 2001.

[8] Chang, P.-H., Park, J., and Park, J.-Y., "Commandless Input Shaping Technique," in *American Control Conference*, (Arlington, VA), pp. 299–304, 2001.

[9] Dharne, A. G. and Jayasuriya, S., "Increasing the Robustness of the Input-Shaping Method using Adaptive Control," in *American Control Conference*, (Denver, Colorado), pp. 1578–1583, 2003.

[10] Drapeau, V. and Wang, D., "Verification of a Closed-loop Shaped-input Controller for a Five-bar-linkage Manipulator," in *IEEE Conf. on Robotics and Automation*, (Atlanta, Ga), pp. 216–221, 1993.

[11] Eberhardt, C. and Sarideréli, M. K., "A Novel Formal Loop Shaping Design Procedure," in *32nd Midwest Symposium on Circuits and Systems*, (Champaign, IL), pp. 124–127, 1990.

[12] Fortgang, J. and Singhose, W., "Improving Trajectory Following by Controlling Internal Deflections Through Input Shaping," in *7th International Conference on Motion and Vibration Control*, (St. Louis), 2004.

[13] Fortgang, J., Singhose, W., Marquez, J., and Perez, J., "Command Shaping for Micro-Mills and CNC Controllers," in *2005 American Control Conference*, (Portland, OR), pp. 4531–4536, 2005.

[14] Franklin, G. F., Powell, J. D., and Emami-Naeini, A., *Feedback Control of Dynamic Systems*. Addison-Wesley Publishing Company, Inc., 3rd ed., 1994.

[15] Gimpel, D. and Calvert, J., "Signal Component Control," *Transactions of the AIEE*, vol. 71, part II, pp. 339–343, 1952.

[16] Gopalakrishnan, J., Reddy, K., and Singh, T., "Concurrent Feedback/Feedforward Design for Second Order Systems," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (San Francisco, CA), 2005.

[17] Gorinevsky, D. and Vukovich, G., "Nonlinear Input Shaping Control of Flexible Spacecraft Reorientation Maneuver," *J. of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 264–270, 1998.

[18] Grassi, E. and Tsakalis, K., "PID controller tuning by frequency loop-shaping," vol. vol.4 of *Proceedings of the 35th IEEE Conference on Decision and Control (Cat. No.96CH35989)*, (Kobe, Japan), p. 4776, IEEE, 1996.

[19] Grassi, E., Tsakalis, K., Dash, S., Gaikwad, S., and Stein, G., "Adaptive/Self-Tuning PID Control by Frequency Loop-Shaping," in *39th IEEE Conference on Decision and Control*, (Sydney, Australia), pp. 1099–1101, 2000.

[20] Grosser, K. E. and Singhose, W., "Command Generation for Reducing Perceived Lag in Flexible Telerobotic Arms," *JSME International Journal*, vol. 43, no. 3, pp. 755–761, 2000.

[21] Haddad, W. M., Kapila, V., and Abdallah, C. T., "Stabilization of Linear and Nonlinear Systems with Time Delay," in *American Control Conference*, (Alburquerque, NM), pp. 3220–3224, 1997.

[22] Harris, E. L. and Smith, O. J. M., "Novel Circuit Damps Transients in Voice-Operated Transmitters," *Electronics*, vol. 35, no. 39, pp. 66–67, 1962.

[23] Hyde, J. M. and Seering, W. P., "Using Input Command Pre-Shaping to Suppress Multiple Mode Vibration," in *IEEE Int. Conf. on Robotics and Automation*, vol. 3, (Sacramento, CA), pp. 2604–2609, IEEE, 1991.

[24] Jones, S. and Ulsoy, A. G., "An Approach to Control Input Shaping with Application to Coordinate Measuring Machines," *J. of Dynamics, Measurement, and Control*, vol. 121, no. June, pp. 242–247, 1999.

[25] Kang, M. and Yang, B., "Discrete Time Noncollocated Control of Flexible Mechanical Systems Using Time Delay," *Journal of Dynamic Systems, Measurement, and Control*, vol. 116, pp. 216–222, 1994.

[26] Kao, C. K., Sinha, A., and Mahalanabis, A. K., "A Digital Algorithm for Near-Minimum-Time Control of Robot Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 320–327, 1987.

[27] Kapila, V., Haddad, W. M., and Grivas, A., "Stabilization of linear systems with simultaneous state, actuation, and measurement delays," *International Journal of Control*, vol. 72, no. 18, pp. 1619–1629, 1999.

[28] Kapila, V., Tzes, A., and Yan, Q., "Closed-Loop Input Shaping for Flexible Structures using Time-Delay Control," in *38th Conference on Decision and Control*, (Phoenix, Az), pp. 1561–1566, 1999.

[29] Kapila, V., Tzes, A., and Yan, Q., "Closed-Loop Input Shaping for Flexible Structures Using Time-Delay Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 454–460, 2000.

[30] Kenison, M. and Singhose, W., "Concurrent Design of Input Shaping and Feedback Control for Insensitivity to Parameter Variations," in *Sixth Int. Workshop on Advanced Motion Control*, (Nagoya, Japan), 2000.

[31] Kenison, M. and Singhose, W., "Concurrent Design of Input Shaping and Proportional Plus Derivative Feedback Control," *ASME J. of Dynamic Systems, Measurement, and Control*, no. September, 2002.

[32] Khalid, A., Singhose, W., Huey, J., and Lawrence, J., "Study of Operator Behavior, Learning, and Performance Using an Input-Shaped Bridge Crane," in *CCA/ISIC/CACSD*, (Taipei, Taiwan), 2004.

[33] Khalid, A., Singhose, W., Huey, J., Lawrence, J., and Frakes, D., "Human Operator Performance Testing Using an Input-Shaped Bridge Crane," *Accepted for Publication in Journal of Dynamic Systems, Measurement, and Control*, 2005.

[34] Kozak, K., Ebert-Uphoff, I., and Singhose, W., "Locally Linearized Dynamic Analysis of Parallel Manipulators and Application of Input Shaping to Reduce Vibrations," *ASME J. of Mechanical Design*, vol. 126, pp. 156–168, 2004.

[35] Kumar, R., "Effects of Time Delays on the Stability of Collocated and Noncollocated Point Control of Discrete Dynamic Structural Systems," *Applied Mathematics and Computation*, vol. 70, pp. 215–232, 1995.

[36] Kwon, D.-S., Hwang, D.-H., Babcock, S. M., and Burks, B. L., "Input Shaping Filter Methods for the Control of Structurally Flexible, Long-Reach

Manipulators," in *IEEE Conf. on Robotics and Automation*, vol. 4, (San Diego, CA), pp. 3259–64, 1994.

[37] Lawrence, J., Fatkin, B., Singhose, W., Huey, J., Weiss, R., Erb, A., and Glauser, U., "An Internet-Driven Tower Crane for Dynamics and Controls Education," in *7th IFAC Symposium on Advances in Control Education*, (Madrid, Spain), 2006.

[38] Lawrence, J., Falkenberg, M., Singhose, W., and Pelaez, G., "Input Shaping for a Flexible, Nonlinear, One-Link Robotic Arm with Backlash," in *2004 USA-JAPAN Symposium on Flexible Automation*, (Denver, Colorado), ASME, 2004.

[39] Lawrence, J. and Singhose, W., "Design of a Minicrane for Education and Research," in *6th International Workshop on Research and Education in Mechatronics*, (Annecy, France), pp. 254–259, 2005.

[40] Lawrence, J., Singhose, W., and Heckman, K., "Robust Friction-Compensating Input Shapers," in *8th Cairo Univ. Int. Conf. on Mechanical Design and Production*, (Cairo, Egypt), 2004.

[41] Lawrence, J., Singhose, W., and Hekman, K., "An Analytical Solution for a Zero Vibration Input Shaper for Systems with Coulomb Friction," *American Control Conference*, 2002.

[42] Lee, E., Park, J., Loparo, K. A., Schrader, C. B., and Chang, P. H., "Bang-Bang Impact Control Using Hybrid Impedance/Time-Delay Control," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 2, pp. 272–277, 2003.

[43] Lewis, D., Parker, G. G., Driessen, B., and Robinett, R. D., "Command Shaping Control of an Operator-in-the-Loop Boom Crane," in *American Control Confernce*, (Philadelphia, Pennsylvania), pp. 2643–2647, 1998.

[44] Liu, Q. and Wie, B., "Robust Time-Optimal Control of Uncertain Flexible Spacecraft," *J. of Guidance, Control, and Dynamics*, vol. 15, no. 3, pp. 597–604, 1992.

[45] Magee, D. and Book, W., "The Application of Input Shaping to a System with Varying Parameters," in *Japan/USA Symposium on Flexible Automation*, pp. pp. 519–25, 1992.

[46] Magee, D. P. and Book, W. J., "Experimental Verification of Modified Command Shaping Using a Flexible Manipulator," in *Proceedings of the 1st International Conference on Motion and Vibration Control*, (Yokohama, Japan), 1992.

[47] Magee, D. P. and Book, W. J., "Filtering Schilling Manipulator Commands to Prevent Flexible Structure Vibration," in *American Control Conference*, (Baltimore, Md), pp. 2538–2542, 1994.

[48] Magee, D. P. and Book, W. J., "Filtering Micro-Manipulator Wrist Commands to Prevent Flexible Base Motion," in *American Control Conf.*, (Seattle, WA), pp. 924–928, 1995.

[49] Magee, D. P., Cannon, D. W., and Book, W. J., "Combined Command Shaping and Inertial Damping for Flexure Control," in *American Control Conference*, (Albuquerque, NM), pp. 1330–1334, 1997.

[50] Mahmoud, M. S. and Al-Muthairi, N. F., "Design of Robust Controllers for Time-Delay Systems," *IEEE Transactions on Automatic Control*, vol. 39, no. 5, pp. 995–999, 1994.

[51] Mahmoud, M. S. and Bingulac, S., "Robust Design of Stabilizing Controllers for Interconnected Time-delay Systems," *Automatica*, vol. 34, no. 6, pp. 795–800, 1998.

[52] Muenchhof, M. and Singh, T., "Concurrent Feed-Forward/Feed-Back Design for Flexible Structures," in *AIAA Guidance, Navigation and Control Conference*, (Monterey, Ca.), 2002.

[53] Niculescu, S.-I., Fu, M., and Li, H., "Delay-Dependant Closed-Loop Stability of Linear Systems with Input Delay: An LMI Approach," in *36th Conference on Decision and Control*, (San Diego), pp. 1623–1628, 1997.

[54] Nishioka, K., Adachi, N., and Takeuchi, K., "Simple pivoting algorithm for root-locus method of linear systems with delay," *International Journal of Control*, vol. 53, no. 4, pp. 951–966, 1991.

[55] Ogata, K., *System Dynamics*. Upper Saddle River, New Jersey: Pearson Prentice Hall, 3 ed., 1998.

[56] Pao, L. and La-orpacharapan, C., "Shaped Time-Optimal Feedback Controllers for Flexible Structures," *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, pp. 173–186, 2004.

[57] Pao, L. Y. and Lau, M. A., "Robust Input Shaper Control Design for Parameter Variations in Flexible Structures," *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 63–70, 2000.

[58] Pao, L. Y. and Singhose, W. E., "Verifying Robust Time-Optimal Commands for Multi-Mode Flexible Spacecraft," *AIAA J. of Guidance, Control, and Dynamics*, vol. 20, no. 4, pp. 831–833, 1997.

[59] Park, J.-Y. and Chang, P. H., "Vibration Control of a Telescopic Handler Using Time Delay Control and Commandless Input Shaping Technique," *Control Engineering Practice*, vol. 12, pp. 769–780, 2004.

[60] Park, J. and Chang, P.-H., "Learning Input Shaping Technique for Non-LTI Systems," in *American Control Conference*, (Philadelphia), pp. 2652–2656, 1998.

[61] Park, J. and Chang, P.-H., "Learning Input Shaping Technique for Non-LTI Systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, pp. 288–293, 2001.

[62] Rogers, K. and Seering, W. P., "Input Shaping for Limiting Loads and Vibration in Systems with On-Off Actuators," in *AIAA Guidance, Navigation, and Control Conference*, (San Diego, CA), 1996.

[63] Seth, N., Rattan, K., and Brandstetter, R., "Vibration Control of a Coordinate Measuring Machine," in *IEEE Conf. on Control Apps.*, (Dayton, OH), pp. 368–73, 1992.

[64] Singer, N., Singhose, W., and Kriikku, E., "An Input Shaping Controller Enabling Cranes to Move Without Sway," in *ANS 7th Topical Meeting on Robotics and Remote Systems*, vol. 1, (Augusta, GA), pp. 225–231, 1997.

[65] Singer, N. C. and Seering, W. P., "Preshaping Command Inputs to Reduce System Vibration," *J. of Dynamic Sys., Measurement, and Control*, vol. 112, no. March, pp. 76–82, 1990.

[66] Singh, T. and Vadali, S. R., "Robust Time-Optimal Control: A Frequency Domain Approach," *J. of Guidance, Control and Dynamics*, vol. 17, no. 2, pp. 346–353, 1994.

[67] Singhose, W. and Singer, N., "Effects of Input Shaping on Two-Dimensional Trajectory Following," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 881–887, 1996.

[68] Singhose, W., Banerjee, A. K., and Seering, W., "Slewing Flexible Spacecraft with Deflection-Limiting Input Shaping," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 2, pp. 291–298, 1997.

[69] Singhose, W., Biediger, E. O., and Chen, Y.-H., "Reference Command Shaping Using Specified-Negative-Amplitude Input Shapers for Vibration Reduction," *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, pp. 210–214, 2004.

[70] Singhose, W., Porter, L., Kenison, M., and Kriikku, E., "Effects of Hoisting on the Input Shaping Control of Gantry Cranes," *Control Engineering Practice*, vol. 8, no. 10, pp. 1159–1165, 2000.

[71] Singhose, W., Porter, L. J., and Seering, W., "Input Shaped Control of a Planar Gantry Crane with Hoisting," in *American Control Conference*, (Albuquerque, NM), pp. 97–100, 1997.

[72] Singhose, W., Seering, W., and Singer, N., "Residual Vibration Reduction Using Vector Diagrams to Generate Shaped Inputs," *J. of Mechanical Design*, vol. 116, no. June, pp. 654–659, 1994.

[73] Singhose, W., Seering, W., and Singer, N., "The Effect of Input Shaping on Coordinate Measuring Machine Repeatability," in *IFToMM World Congress on the Theory of Machines and Mechanisms*, 1995.

[74] Singhose, W., Singer, N., and Seering, W., "Improving Repeatability of Coordinate Measuring Machines with Shaped Command Signals," *Precision Engineering*, vol. 18, no. April, pp. 138–146, 1996.

[75] Singhose, W., Singer, N., and Seering, W., "Time-Optimal Negative Input Shapers," *J. of Dynamic Systems, Measurement, and Control*, vol. 119, no. June, pp. 198–205, 1997.

[76] Singhose, W., Singh, T., and Seering, W., "On-Off Control of Flexible Spacecraft with Specified Fuel Usage," in *American Control Conference*, (Albuquerque), pp. 2308–2312, 1997.

[77] Singhose, W., Singh, T., and Seering, W., "On-Off Control With Specified Fuel Usage," *Journal of Dynamic Systems, Measurement, and Control*, vol. 121, pp. 206–212, 1999.

[78] Singhose, W. E., Crain, E. A., and Seering, W. P., "Convolved and Simultaneous Two-Mode Input Shapers," *IEE Control Theory and Applications*, no. Nov., pp. 515–520, 1997.

[79] Singhose, W. E., Porter, L. J., Tuttle, T. D., and Singer, N. C., "Vibration Reduction Using Multi-Hump Input Shapers," *J. of Dynamic Systems, Measurement, and Control*, vol. 119, no. June, pp. 320–326, 1997.

[80] Smith, A. and Lee, K.-M., "An Experimental Investigation of Two Digital Tracking Controllers: Adaptive and Non-Adaptive," in *Winter Annual Meeting of the American Society of Mechanical Engineers*, (Dallas, Tx), pp. 9–14, 1990.

[81] Smith, J., Kozak, K., and Singhose, W., "Input Shaping for a Simple Non-Linear System," in *Accepted to the 2002 American Control Conference*, (Anchorage, AK), 2002.

[82] Smith, O. J. M., "Direct Synthesis Through Block Diagram Substitutions," *Convention Record of the IRE (Automatic Control)*, vol. 5, no. 4, pp. 8–16, 1957.

[83] Smith, O. J. M., "Mixed Distributed and Lumped Parameter Systems," in *IRE Wescon Convention Records*, vol. 1, pp. 122–132, 1957.

[84] Smith, O., *Feedback Control Systems*. New York: McGraw-Hill Book Co., Inc., 1958.

[85] Smith, O. J. M., "Posicast Control of Damped Oscillatory Systems," *Proceedings of the IRE*, vol. 45, no. September, pp. 1249–1255, 1957.

[86] Smith, S. C. and Messner, W., "Loop shaping with closed-loop magnitude contours on the bode plot," vol. 4 of *Proceedings of the American Control Conference*, (Anchorage, AK, United States), p. 2747, Institute of Electrical and Electronics Engineers Inc., 2002.

[87] Sorensen, K., *A Combined Feedback and Command Shaping Controller for Improving Positioning and Reducing Cable Sway in Cranes.* Masters Thesis, Georgia Institute of Technology, 2005.

[88] Sorensen, K., Singhose, W., and Dickerson, S., "A Controller Enabling Precise Positioning and Sway Reduction in Cranes with On-Off Actuation," in *IFAC*, (Prague), 2005.

[89] Sorensen, K., Singhose, W., and Dickerson, S., "A Controller Enabling Precise Positioning and Sway Reduction in Bridge and Gantry Cranes," *Accepted for Publication in Control Engineering Practice*, 2006.

[90] Sparks, A., Banda, S., and Yeh, H. H., "A comparison of loop shaping techniques," Proceedings of the 30th IEEE Conference on Decision and Control (Cat. No.91CH3076-7), (Brighton, UK), p. 235, IEEE, 1991.

[91] Staehlin, U. and Singh, T., "Design of Closed-Loop Input Shaping Controllers," in *American Control Conference*, (Denver, Co), pp. 5167–5172, 2003.

[92] Stefano, M.-L., "Optimal Design of PID Regulators," *International Journal of Control*, vol. 33, no. 4, pp. 601–616, 1981.

[93] Swisher, G. and Huang, A., "Posicast Control of Third- and Fourth-Order Feedback Control Systems," in *Annual Pittsburgh Conference on Modeling and Simulation*, (Pittsburgh, PA), pp. 721–726, 1980.

[94] Sze, T. W. and Calvert, J. F., "Short-Time Memory Devices in Closed-Loop System – Steady-State Response," in *AIEE Fall General Meeting*, (Chicago, Ill.), pp. 340–344, 1955.

[95] Tomizuka, M., "Zero Phase Error Tracking Algorithm for Digital Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 65–68, 1987.

[96] Tsao, T.-C. and Tomizuka, M., "Adaptive Zero Phase Error Tracking Algorithm for Digital Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 349–354, 1987.

[97] Tuttle, T. D. and Seering, W. P., "Experimental Verification of Vibration Reduction in Flexible Spacecraft Using Input Shaping," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 4, pp. 658–664, 1997.

[98] Tuttle, T. D. and Seering, W. P., "A Zero-placement Technique for Designing Shaped Inputs to Suppress Multiple-mode Vibration," in *American Control Conf.*, vol. 3, (Baltimore, MD), pp. 2533–2537, 1994.

[99] Tzes, A., "Robust Control Design Issues for Input-Shaped Discrete Systems," in *American Control Conference*, (Denver, Co), pp. 4785–4787, 2003.

[100] Tzes, A. and Yurkovich, S., "An Adaptive Input Shaping Control Scheme for Vibration Suppression in Slewing Flexible Structures," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 2, pp. 114–121, 1993.

[101] Udwadia, F. E., "Noncollocated Control of Distributed-Parameter Nondispersive Systems with Tip Inertias Using Time Delays," *Applied Mathematics and Computation*, pp. 47–75, 1992.

[102] Vescovo, D. D. and D'Ambrogio, W., "Control of a Flexible Link by Shaping the Closed Loop Frequency Response Function Through Optimised Feedback Filters," *Mechanical Systems and Signal Processing*, vol. 9, no. 1, pp. 1–13, 1995.

[103] Wen, J. and Potsaid, B., "An Experimental Study of a High Performance Motion Control System," in *American Control Conference*, (Boston, MA), pp. 5158–5163, 2004.

[104] Zhong, Q.-C. and Hang, C.-C., "Control of Processes with Dead Time and Input Constraints Using Control Signal Shaping," *Control Theory Applications*, vol. 151, no. 4, pp. 473–480, 2004.

[105] Zuo, K., Drapeau, V., and Wang, D., "Closed Loop Shaped-Input Strategies for Flexible Robots," *The International Journal of Robotics Research*, vol. 14, no. 5, pp. 510–529, 1995.

[106] Zuo, K. and Wang, D., "Closed Loop Shaped-Input Control of a Class of Manipulators with a Single Flexible Link," in *IEEE International Conference on Robotics and Automation*, (Nice, France), pp. 782–787, 1992.