Using Input Shaping to Minimize Residual Vibration in Flexible Space Structures

by

Kristen Andrea Bohlke

B.S.E., Mechanical Engineering
Princeton University, 1993

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

**Master of Science
in Mechanical Engineering**

at the

**Massachusetts Institute of Technology
June 1995**

Signature of Author ______________________________
Department of Mechanical Engineering
June 5, 1995

Certified by ______________________________
Professor Warren P. Seering
Thesis Advisor

Accepted by ______________________________
Ain A. Sonin
Chairman, Departmental Graduate Committee

Using Input Shaping to Minimize Residual Vibration in Flexible Space Structures

by

Kristen Andrea Bohlke

Submitted to the Department of Mechanical Engineering
on June 5, 1995 in partial fulfillment of the requirements for the Degree of
Master of Science in Mechanical Engineering

Abstract

Input shaping is examined as a technique to reduce the residual vibration in flexible space structures. As part of a NASA program, a nonlinear three-dimensional model of the Shuttle's Remote Manipulator System (SRMS) was used to test the interactions between feedback and feedforward control techniques. The problem of changing geometry systems was also examined in detail. As these systems move, their geometries change, and thus the system characteristics change. This often leads to problems with controlling such a structure. If a feedback controller is optimized for certain frequencies, then a large move encompassing many frequencies might lead to more residual vibration than usual. These systems also pose interesting problems for input shaping. An input shaper is usually designed for one main frequency. If that frequency is shifting during the move, what frequency do you shape for? This thesis addresses this problem.

Results from using input shaping with feedback controllers on the SRMS show that input shaping does improve performance for small slews over just feedback control alone. For longer slews using a trapezoidal velocity profile, input shaping only helps if the settling criterion is very small. Runs done using the Draper Remote Simulation showed that the response of the system depended very heavily on what type of payload was being moved. For the unloaded case, the SRMS settled very quickly and a quick, insensitive input shaper improved the performance the most. For a midsize payload, the SRMS frequencies are much lower and performance is thus degraded. A more insensitive input shaper gave the best performance increase here. For a very large payload, the system frequencies are around 0.4 Hz. The unshaped SRMS response settled very slowly, but input shaping could not overcome the nonlinearities and did not provide any performance increase.

Thesis Supervisor: Warren Seering
Title: Professor of Mechanical Engineering

Acknowledgments

I would like to thank many people without whom this thesis would not have been possible.

My thesis advisor, Warren Seering, has always been helpful and available. He provided the problems that I tried to solve and the insight to help me see what was wrong with my solutions. He also provided the necessary "kick" to get me to realize that I actually needed to finish my thesis this year. Thank you Warren for helping me learn all that I needed to know.

My compatriots in the Input Shaping field, Tim Tuttle and Bill Singhose, were also wonderful. They were always willing to take the time to explain different aspects of the field to me.

The other members of the FLEX program, Carl Blaurock, Ketao Liu, and Dave Miller, were always ready to explain things to a clueless master's student. I really appreciated that! I also would like to thank the Space Engineering Research Center for all of its help and support.

My friends were the ones who kept me sane throughout the two years, though especially during that last dreadful spring. Thanks Karen, Susan, Amy, Sandeep, Barbara, Kevin, Sara, and Helen! Q kept me company in the office during the last two years and was always ready to take a break and surf the Web.

My parents and brother have always been supportive during all of my endeavors. They were always ready with encouraging suggestions and were always there just to listen. Thanks for everything you have done. I think after years of schooling and lectures from you, I am now finally ready to enter the real world.

This work describes research performed at the MIT Space Engineering Research Center. Funding for this research was provided by the National Aeronautics and Space Administration under grant #NAGW-1335 in conjunction with the SERC Training Grant, #NGT-10032.

Table of Contents

List of Figures

List of Tables

Introduction

Chapter 1

1.1 Background and Motivation

Vibrations exist all around us. From the motion of a car as it travels over a series of bumps to the vibration of the tip of a manufacturing robotic arm to the spinning of a computer's hard drive, vibrations are a part of everyday life. And many common engineering problems involve getting rid of or reducing the levels of vibration. It is possible to reduce vibration by adding stiffness to the system, but that often adds cost and slows down the operation speed. A better solution is to intelligently choose a control strategy that minimizes residual vibration, yet still moves quickly.

Vibration can be an especially large problem when working with space structures. For example, the astronauts who control the Shuttle's Remote Manipulator System (SRMS) often encounter significant time delays because they must wait for vibrations to die out before they can attempt accurate positioning tasks. This problem is especially acute because the SRMS's fundamental frequency is so low, around 0.1 Hz when carrying a midsized payload. If the astronauts must wait 10 cycles for the vibrations to die down, there will be a delay of 100 seconds. A method of reducing the vibration and reducing the waiting time would save time and money.

Many different techniques have been developed to reduce vibrations. The feedback control field has worked for many decades to discover and develop better and more robust kinds of feedback controllers. Controllers are now capable of dealing with multiple inputs, multiple outputs, many sensors, and actuators. Due to their ability to adjust to disturbances, closed-loop controllers are often indispensable for insuring adequate performance. Feedback control is able to reduce the residual vibration by increasing the closed-loop damping ratio. However, the amount of damping that can be added is often limited by design constraints. Active vibration control of flexible structures, such as flexible robotic manipulator systems, has also experienced rapid growth in recent years. The technique has been focused on eliminating vibrations that result in the structure when feedback control is applied. Not as much attention has been paid to the idea of modifying the system input so that the vibration is not excited in the first place.

Input shaping is one such scheme that works upon the principle of modifying the system input by taking out the energy at the system frequencies. Theses frequencies are not excited by the input and thus do not oscillate. Input shaping was first developed to work with linear systems, but can easily be applied to nonlinear systems as well. The only knowledge necessary is the approximate system frequencies and damping ratios. These frequencies and dampings are used to create a shaped input that does not contain energy at the specified frequencies. The cost of using input shaping is a time delay; the shaped input ends after the unshaped input. However, this is usually an acceptable tradeoff because the system is oscillating long after the command is over anyway. The input shaper delays the command, but the shaped system response still settles before the unshaped response.

1.2 Previous Work in Input Shaping

The origins of input shaping can be traced to Smith and his idea of posicast control in 1958. [29] His idea applies to one-mode systems and involves breaking a step command into two smaller steps, one of which is delayed in time. This shaped command results in a reduced settling time of the response. However, the posicast method is not very robust, since the system must have only one vibrating mode and the frequency must be known exactly.

The first person to fully realize input shaping's potential was Singer. [25] In his doctoral thesis, he derived the mathematical equations behind input shaping and provided the tools for generating impulse sequences for many different kinds of systems. By extending the field of input shaping to cover systems with various dampings and multiple modes, Singer made input shaping into a viable vibration reduction technique. [24] contains a concise summary of the mathematics and the implementation of input shaping.

Singhose further extended the field by his derivation of increased insensitivity input shapers. [28] This advance was made possible by relaxing the zero vibration constraint at the system's natural frequency. By allowing the residual vibration to be some nominal level, the insensitivity curve widens around the frequency and is less sensitive to modeling errors. Singhose and Singer also worked on the time-optimal negative input shapers. [27] Traditionally, the input shaper has contained only positive amplitude impulses. However, when the impulses are allowed to have negative amplitudes, the length of the shaper can be greatly reduced. The negative shapers are slightly less insensitive than the comparable positive shapers. A nice comparison of input shaping and filtering techniques appears in [26].

Hyde calculated direct solutions to the multiple mode problem. [11] He reformulated Singer's single mode shaper equations to include several modes and solved the equations simultaneously using an optimizing software program. Hyde also applied the input shapers to an experimental flexible structure, the

MACE testbed. Chang worked with the same hardware and did more tests with different controllers with varying bandwidths. He saw that a higher bandwidth controller and input shaper had improved percentage vibration reduction and the absolute least residual vibration compared to lower bandwidth controllers and input shapers. [5]

By transforming input shaper design into the z-plane, Tuttle developed a different way of deriving multiple-mode input shapers. [32] Zero-placement in the z-plane provides great flexibility in shaper design that can be exploited to improve performance. For example, time-optimal sequences are easily generated if the digital sampling rate is low.

Input shaping has been applied to many different types of system. Jones and Ulsoy use an input shaper to avoid exciting unwanted vibrations in a Coordinate Measuring Machine. [12] Tzes, Englehart, and Yurkovich apply input shaping to a flexible one-link manipulator and achieve good performance. [31] Input shaping has been implemented on a spherical pointing motor to reduce oscillations by a factor of ten. [3] Input shaping has also been applied to wafer handling robots, disk drives, a heavy-lift hydraulic robot, and a wafer stepper used to manufacture microchips. Magee and Book use input shaping on a flexible arm test bed with an attached Schilling micro-manipulator. [14] Banerjee applied input shaping to a nonlinearly elastic shuttle antenna, where the shuttle was constrained to use only bang-bang inputs. [2]

1.3 Previous Work on Flexible Robotic Systems

The Flexbot, a three-degree-of-freedom flexible system, was designed by Christian to closely resemble the first three joints of the SRMS. [6] Christian tested a variety of trajectories designed to minimized residual vibration. He found that a trapezoidal velocity profile combined with input shaping is the best method for eliminating vibration without sacrificing overall move time. Rappole implemented an adaptive method of input shaping on the Flexbot and compared it to constant-valued input shapers. The adaptive shapers did not perform as well as the constant input shapers for constant-frequency moves, but showed promise in reducing vibrations in systems with large frequency variations. [20]

Meckl and Kinceler investigated a two-link robot with flexible joints for a large angle trajectory move. [15] They derived optimal minimum-energy acceleration profiles for this model and got very good results in a preliminary simulation. Magee and Book work with a two-link, flexible manipulator and use it to test a modified command filtering methods. [13] They implement input shaping inside the closed-loop system and compare it with a modified command filtering method for a system whose parameters vary with time.

Schmitz and Ramey used a long-reach, 3-DOF planar manipulator to compare a colocated independent joint control design and an end-point position sensor feedback controller. [22] The end-point sensing was implemented with a photodetector; however a wrist-mounted CCD camera is a more realistic option for space systems. Tzes and Yurkovich worked with a single, very flexible link and large slewing moves and applied an adaptive shaping technique which incorporated frequency identification. [30] Carusone and D'Eleuterio work with a two-link planar manipulator with rotary joints supported by air pucks on a flat horizontal table, with a variety of rigid and flexible links. [4]

Oakley and Cannon use a two-link flexible manipulator to test modern feedback control techniques such as an LQG-based endpoint controller, which is noncolocated. [18] In this case, the inner link is rigid and the outer link is flexible. The LQG-EP controllers worked much better than a PID controller, with no significant increase in torque requirements. Hollars and Cannon did experiments on a two-link manipulator with flexible tendons. [10] They investigated classically designed colocated control and modern state-space noncolocated control; the noncolocated controller had better performance.

Hillsley and Yurkovich apply input shaping to a two-link flexible, planar manipulator, with and without feedback control. [9] The use of impulse shaping with an endpoint-feedback controller provides superior performance over each technique alone. Feddema uses a infinite impulse response filtering technique to reduce vibration in a two-link flexible arm and a gantry crane with a suspended payload. [8] Zuo and Wang implement a closed-loop input shaper on a single flexible link and achieve good vibration control and stability, even when disturbances are introduced. [34] Drapeau and Wang present a closed-loop shaped-input control strategy implemented on a five-bar linkage manipulator with one flexible beam. [7]

1.4 Outline

The remainder of this thesis is divided into five chapters. Chapter 2 contains some interesting applications of input shaping theory. The interactions between damping ratios and different types of input shapers are examined. The performance of the input shapers for various modeling errors and settling bands is explained and a strategy is recommended. Another sub-problem that is investigated is the effect of friction on input shaping.

Chapter 3 explores the FLEX program and its components. The FLEX program is a NASA In-Step program whose purpose is to develop the best way of controlling a remote manipulator in space. The space arm will ultimately be used to construct the space station, so precise and fast positioning is essential. A team from MIT, Martin Marietta, Convolve, and Payload Systems was assembled for Phase A. Further details of the background and motivation of the project will be given in Chapter 3.

Chapter 4 gives some results from Phase A of the FLEX program. Several different models and workspaces were explored during Phase A. An unloaded model was developed and tested in various configurations. A study was done on the interactions between input shaping and feedback controllers. A midsized payload model was tested to see how different configurations and move durations affected the relationship between feedback controllers of varying complexity and inputs shaping.

Chapter 5 explores the changing geometry systems problem. In previous work on input shaping, certain areas have been briefly touched upon, but not delved into. One of these areas is the problem of changing geometry systems. As these systems move, their geometries change, and thus the system characteristics change. This often leads to problems with controlling such a structure. If a feedback controller is optimized for certain frequencies, then a large move encompassing many frequencies might lead to more residual vibration than usual. These systems also pose interesting problems for input shaping. An input shaper is usually designed for one main frequency. If that frequency is shifting during the move, what frequency do you shape for? Chapter 5 will attempt to address this problem.

Chapter 6 concludes the thesis with an overview of the results and a suggestion of future work.

Input Shaping Studies

Chapter 2

2.1 Introduction

Input shaping involves convolving a sequence of impulses, otherwise known as the input shaper, with a desired system command to produce the shaped system command. The input shaper is calculated to eliminate vibrations at certain desired frequencies. It can be made insensitive to variation in resonant frequencies, and thus is more effective at minimizing vibration in flexible systems whose frequencies shift during moves. This chapter presents an overview of the implementation of input shaping as well as some short studies on various aspects of input shaping.

2.2 Explanation of Input Shaping

Input shaping is a feedforward technique that is implemented outside the feedback loop, as shown in Figure 2.1. The command is generated and then the input shaper is convolved with the command. The input shaper is designed to eliminate or reduce the vibration at certain frequencies, usually the important system modal frequencies. Input shaping's most straightforward form uses a simple algorithm developed by Singer to reduce the residual vibration in flexible systems. Singer derived the technique from a second-order, linear model of a vibrating system. The equations are given at length in several references, so I will not present them here. [11, 25]

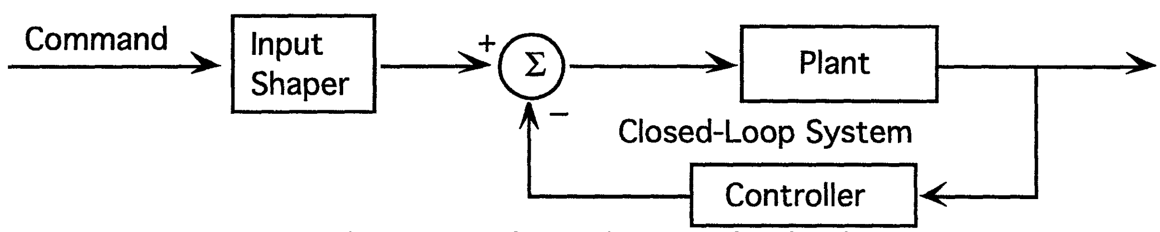


Figure 2.1: Implementation of input shaping with a closed-loop system

Input shaping is calculated as follows. A series of impulses is specified such that the response of a second-order system satisfies various constraints. The constraints include the following: the residual vibration at the end of the move must be zero, the first impulse occurs at t=0, and the amplitudes of the impulses must sum to one. The derivative of the residual vibration can also be set equal to zero, which gives the system additional constraints and more insensitivity to modeling error. After solving the equations, the result is a series of impulses, each with an amplitude and time. For example, a zero vibration (ZV) input shaper only has four constraints, and thus two impulses. A zero vibration, zero derivative (ZVD) shaper has six constraints and three impulses. The amplitudes and timing of a ZVD shaper's impulses are given in Equation 2.1.

Each input shaper is generated for a specific frequency and damping ratio. If the exact system frequency and damping are not known, the input shaper can be made more insensitive to modeling errors by adding additional derivative constraints. For example, a ZVDD input shaper has zero residual vibration and two additional derivative constraints. The sensitivity of a shaper to modeling errors can be calculated mathematically. Sensitivity curves are a way to graphically portray the robustness of a particular shaper by showing the residual vibration as a function of frequency. Both axes are normalized to generalize the curve. The normalized frequency is defined as the natural frequency of the system divided by the frequency used to design the input shaper. The percentage of vibration remaining is the residual vibration amplitude with shaping divided by the residual vibration of the unshaped response. Insensitivity of an input shaper is defined as the width of the sensitivity curve at a given level of residual vibration. Vibration levels of 5% and 10% are commonly used to calculate insensitivities.

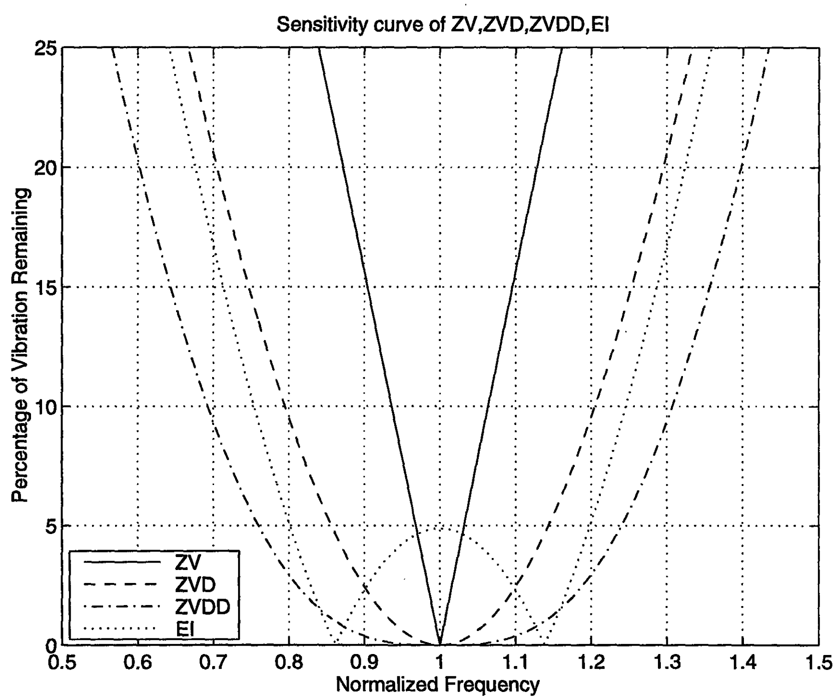


Figure 2.2: Sensitivity curves for different input shapers

Figure 2.2 shows the sensitivity curves that result from different types of input shapers. It is clear that additional derivative constraints widen the sensitivity curve. The extra-insensitive (EI) shaper is as long in duration as the ZVD shaper, but is more insensitive. It gains this insensitivity by relaxing the zero vibration constraint at the modeling frequency. Instead, the residual vibration at that point is limited to some small value, V, and then the zero vibration constraint can be enforced at two frequencies close to the modeling frequency. This leads to the wider sensitivity curve with a hump in the center where the residual vibration reaches the level V. V is usually chosen to be 5 or 10%.

Input shaping is not without its price, however. Once the input shapers are calculated, they are convolved with the system command. The impulses end up delaying the end of the command. Figure 2.3 shows an example of how a command is convolved with an input shaper. ZVD impulses are convolved with a step command and the resulting shaped command is shown. The amplitudes and times of the ZVD input shaper are given by Equation 2.1 and are only a function of frequency (ω) and damping ratio (ζ).

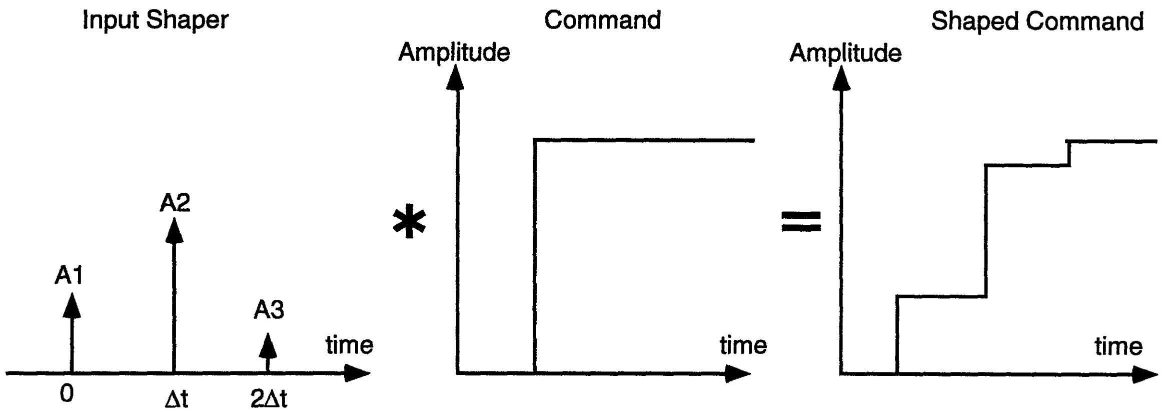


Figure 2.3: Example of convolving a step command with a ZVD shaper

$$K = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \qquad A_1 = 1/D$$

$$D = 1 + 2K + K^2 \qquad A_2 = 2K/D \tag{2.1}$$

$$\Delta t = \frac{\pi}{\omega\sqrt{1-\zeta^2}} \qquad A_3 = K^2/D$$

When the input shaper is convolved with a simple step command, the resulting shaped command is fairly simple to explain. The shaped command starts at t=0 with amplitude A_1. Then at Δt, the second impulse is added to the first, so the amplitude of the step is A_1+A_2. At $2\Delta t$, the final impulse is added and the shaped move catches up with the unshaped move because $A_1+A_2+A_3=1$. The command ends up being delayed $2\Delta t$ seconds. If there is a lot of residual vibration in the unshaped step response, i.e. it lasts longer than $2\Delta t$, then input shaping gets rid of all the residual vibration by the end of the move and time is saved. If the system is highly damped, the vibrations from the unshaped step response might die out before the shaped move even ends, and time is lost. Careful selection of when to use the input shaper is important.

Different input shapers have different time delays associated with them. Usually, the more insensitive the input shaper is, the longer the associated time delay is. Table 2.1 shows the insensitivities and time delays for different input shapers. The % insensitivity is defined as follows: the distance from the center frequency to the point where the sensitivity curve hits 5% residual vibration. So for a ZVD shaper, the system can accept a error in the frequency of ±14% and there will still be less than 5% residual vibration at the end of the shaped move. This can also be seen in Figure 2.2. Negative shapers allow the amplitudes of the impulses to be negative, which can lead to saturation. However, they are much shorter in duration than the other shapers.

type of shaper	% insensitivity	duration of input shaper (cycles)
ZV	±3	0.50
negative ZV	±3	0.29
ZVD	±14	1.00
negative ZVD	±13	0.68
ZVDD	±24	1.50
EI	±20	1.00
negative EI	±18	0.68
EI 2 hump	±36	1.50

Table 2.1: Shapers and their associated insensitivities and delays

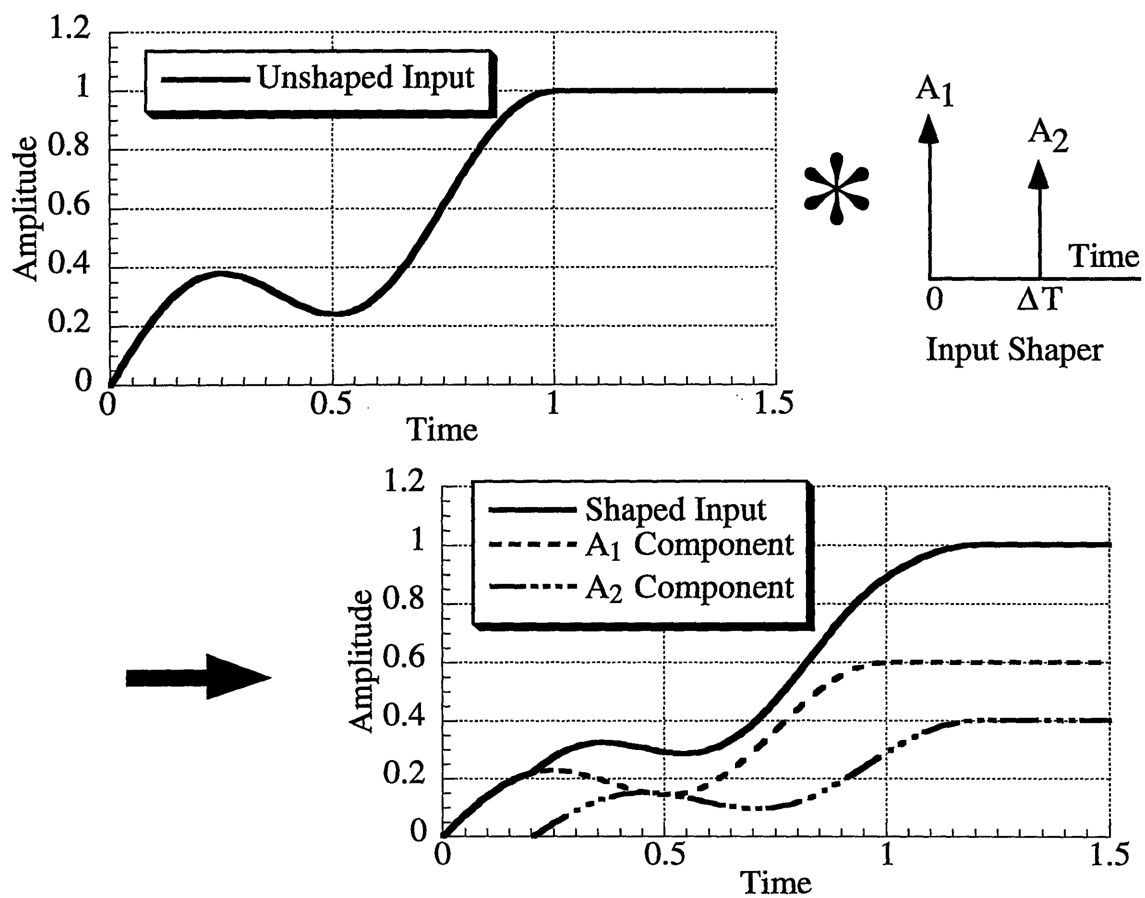


Figure 2.4: Convolution broken down into its components

Figure 2.4 shows the convolution process for a non-step command and a ZV shaper. The resulting shaped command does not look like the original, but the important frequency component has been removed, so there will be less residual vibration at that frequency.

Once these single mode shapers were derived, it was quickly realized that two single mode shapers could be created for different modes and convolved together, thus creating a multi-mode input shaper. Convolution does not create optimally short input shapers unless the frequencies of the two modes are far apart. Optimization programs have been used to find direct solutions for multiple mode problems. Another method of finding an optimally short input shaper involves searching the Z domain. This method is described by Tuttle in [32].

Input shaping is not merely a technique that only works on pre-computed commands. It also can be implemented in real-time and works very well on unknown trajectories. This is one of its strengths; input shaping only requires foreknowledge of the system frequencies and dampings.

2.3 Interactions between the Damping Ratio and Input Shaping

Input shaping is able to reduce the level of vibration by modifying the original command to filter out the energy at certain important frequencies. The price of the reduced vibration is a time delay, which is equal to half of the fundamental frequency's period if a zero vibration (ZV) shaper is used. One of the ways of measuring the amount of residual vibration is to calculate the settling time of the system. The settling time is defined to be the time required for the system's response to a command to reach and stay within a range about the final value. The range is usually two or five percent of the final value. Another way to decrease the residual vibration is to increase the system's damping ratio, either by system modifications or by using a feedback controller. As the system's damping increases, the step response approaches the critically damped case where the response rises very slowly, but has no residual vibration and thus settles during its initial rise. Since input shaping adds a time delay, it seems that above some damping ratio the time delay should make the shaped system response slower than the unshaped case. It was decided to investigate the interactions between damping and input shaping, to see how much time was saved at different damping ratios.

For the single mode case, it is very easy to generate input shapers if the system frequency and damping ratio are known. The main question is how well are the system parameters known? If the system parameters are very well know, a simple ZV shaper, which contains two impulses, will work very well to get rid of the vibration. However, if the system parameters are not known as well, an input shaper with more insensitivity to modeling errors is needed, such as a zero vibration and zero derivative (ZVD) or a ZVDD shaper. The sensitivities of these shapers are shown in Figure 2.2. The ZV shaper is the most sensitive, since a small shift in system frequency causes a large change in the amount of residual vibration. The ZVDDD is the most insensitive, but also has the longest time delay of two periods of vibration for a system with no damping.

To get the additional insensitivity to modeling errors, you must pay the price of an extra time delay. At some damping ratio the tradeoff between shorter settling time and longer command time delay should come into play. At that point it makes sense to change the type of shaper being using, from an insensitive shaper to a more sensitive shaper. This study examined the tradeoffs between system damping, insensitivity of the input shaper, and the amount of modeling error.

2.3.1 System Description

A model of a linear second-order system was created. The physical system is shown in Figure 2.5. The open loop transfer function is given by Equation 2.2. No feedback controller was added to the system, so there is only one vibratory mode. The natural frequency was chosen to be 1 radian/sec and the damping ratio was varied.

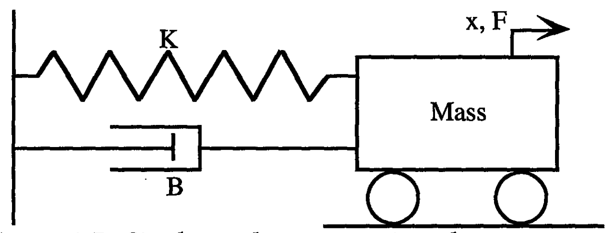


Figure 2.5: Single-mode mass, spring, damper system

$$G(s) = \frac{\omega_n^{\,2}}{s^2 + 2\zeta\omega_n s + \omega_n^{\,2}}$$

$$\omega_n^2 = \frac{K}{m} \quad \zeta = \frac{B}{2\sqrt{mK}} \tag{2.2}$$

MATLAB was used to generate the response of this system to a shaped and unshaped step input. The settling times were calculated for settling bands of 1, 2, and 5 % of the final value. This was done for a range of damping ratios from ζ=0.01 to ζ=0.9 and for the ZV, ZVD, ZVDD, and ZVDDD input shaper cases. The test matrix axes are shown in Table 2.2.

Axes	Range of values
damping	0.01-0.90
system error	2%, 5%, 10%, 20%
input shaper	unshaped, ZV, ZVD, ZVDD, ZVDDD

Table 2.2: Axes of test matrix

Since input shaping cancels all vibration if the system is known perfectly, an error in system knowledge was added to the system model. Runs were done for a 2%, 5%, 10%, and 20% error in the system knowledge. Error in system knowledge was defined by Equation 2.3, where ω_n is the natural frequency of the system and e is the percent error. Thus the input shaper is shaping for a frequency, ω, that is lower than the actual frequency, ω_n.

$$\omega = \frac{\omega_n(100 - e)}{100} \tag{2.3}$$

The data is presented in several graphs comparing the damping ratio to the percentage of time saved by using input shaping. Percentage time saved is defined by Equation 2.4.

$$\%\text{ time saved} = \frac{ts_{unshaped} - ts_{shaped}}{ts_{unshaped}} \tag{2.4}$$

A higher % time savings was better since that implied that the shaped step response settled faster than the unshaped response. This performance metric has an upper value of one and will be negative once the unshaped response settles faster than the shaped response.

2.3.2 Results

Figure 2.6 is for the 2% error and 1% settling band case. In this case, the ZVD shaper saves the most time for dampings below 0.2. This is because the ZVDD rise time is much slower than the ZVD because of the extra half-period delay. Since the settling time is being caught during the initial rise, the length of the shaped command matters. The ZV shaper still has residual vibration after the command is over because the small modeling error affects this very sensitive shaper and causes a residual vibration greater than the 1% settling band. The unshaped settling time decreases as the damping increases and the ZV settling time is keeping pace with that; the straight line reflects this behavior. After ζ=0.2, the ZV shaper saves the most time because the combination of the higher damping ratio and the input shaper reduces the vibration enough that the ZV settles during the initial rise. Above a damping ratio of 0.6, the ZVD and ZVDD shapers actually hurt the performance since the unshaped response rises and settles before the shapers finish rising.

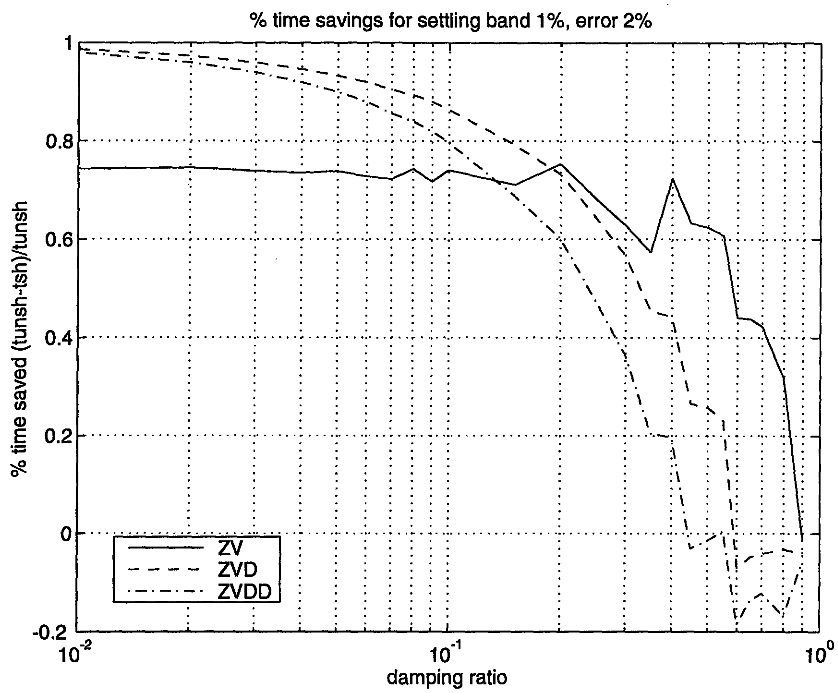


Figure 2.6: Time savings for a 1% settling band, 2% error

The comparable graph for 5% error and a 1% settling band is not very different from Figure 2.6, since there is not a large difference in percent error in system knowledge. The main difference is that in the 5% error and 1% settling band case, the ZV line is higher and crosses the other two lines at a higher damping ratio.

Figure 2.7 and Figure 2.8 show the time histories for ZV shaper with a 5% error in system knowledge and a 5% settling band. The two horizontal lines at y=1.05 and y=.95 are the settling band lines. Figure 2.7 shows the ZV shaper for a low damping ratio, and would correspond to the flat curve of the ZV shown in Figure 2.6. For example, the shaped and unshaped settling times for ζ=.07 are around 8 and 40 respectively for a % time savings ratio of 0.80, and the times for ζ=.10 are 5 and 30 for a % time savings ratio of 0.81. These ratios are comparable and explain the flatness of the curve.

These time histories are for the mass-spring system shown in Figure 2.5. The unshaped response behaves as expected. It overshoots the desired position and then comes to rest at the equilibrium position. The shaped response overshoots the first step of the shaped command and begins to move back towards the first part of the shaped step. When the shaped command changes to the final desired position, the system is pulled back towards the new desired position and comes to rest much more quickly than the unshaped response. If there was no error, the shaped command would end precisely when the system had reached the desired position and there would be no residual vibration. The error in the shaper frequency means that the end of shaped command is delayed too long.

Therefore, the system has already reached the final desired position and changed direction to move toward the first part of the shaped step. A example of a shaped step command is shown in Figure 2.3.

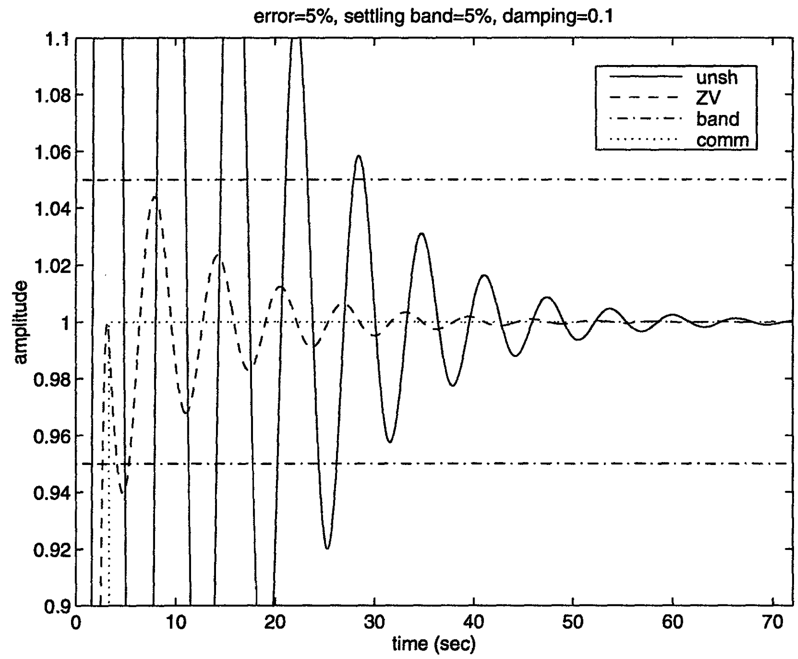


Figure 2.7: Time history for the ZV and unshaped step responses

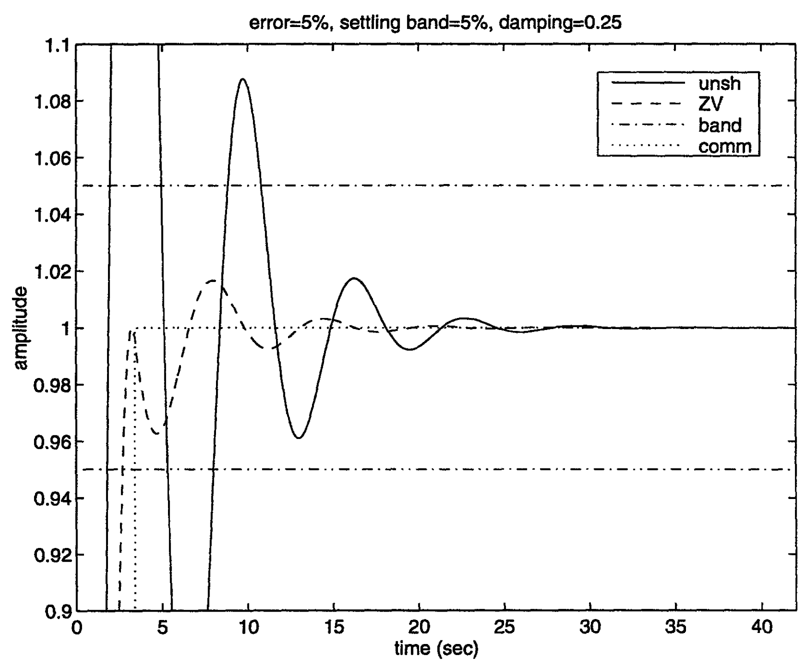


Figure 2.8: Time history for the ZV and unshaped step responses

Figure 2.8 shows the time history for the case where the shaped response settles immediately to within the settling band, before the response is done rising. For these cases, settling time is a function of rise time and the quicker the response rises, the more time it saves. This is why the ZV response can save more time than the ZVD or ZVDD response, even though the ZV allows larger vibration amplitudes. The ZV response gets there before one-half of a period of the system, while the ZVD case settles just before one period and the ZVDD case settles just before 1.5 periods. Meanwhile, the unshaped response is getting slower as damping increases, but also has less overshoot, so settles more quickly. The shaped settling time is staying approximately the same, while the unshaped settling time is decreasing. This leads to the lines seen in Figure 2.6 for the ZVD and ZVDD shaper cases. The shaped settling time is not varying much, while the unshaped is varying.

Figure 2.9 is for the 10% error case with a 1% settling band. Here both the ZV and ZVD shaper curves are flat until ζ=0.2. The ZVDD line is much higher, since it settles to within 1% very quickly by comparison.

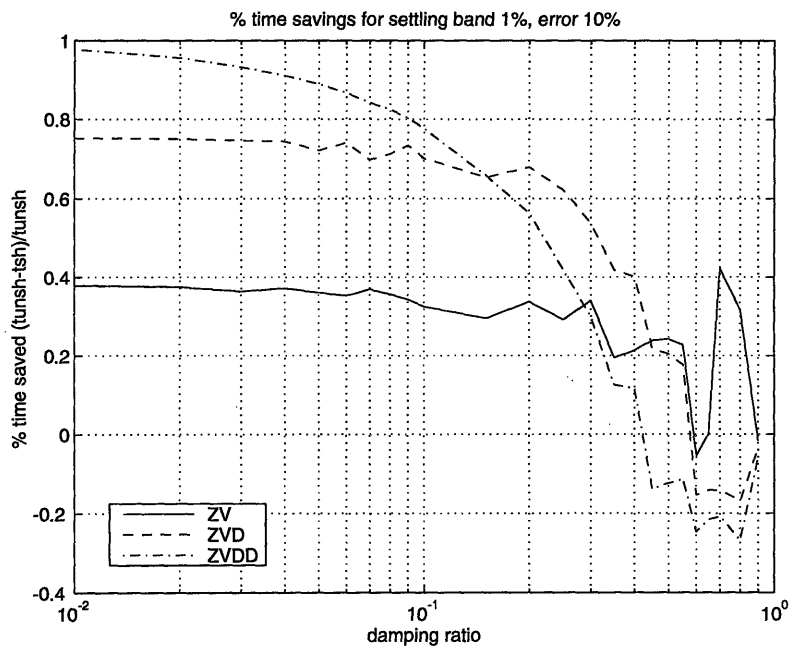


Figure 2.9: Time savings for a 1% settling band and a 10% error

Figure 2.10 shows the 20% system error case with a 1% settling band. All of the shaping cases are flat, so the ZVDD has the most time savings. None of the time savings are as large as seen before, since there is a large error in system knowledge. This reflects the sensitivity curve pictured in Figure 2.2. The ZVDD has the widest base, and so reduces vibration the most even with large modeling errors. Figure 2.11 shows the 2% error case with a 5% settling band. All of the curves are decreasing with increasing damping, which implies that all of the shaped responses are settling during the initial rise. The ZV shaper rises the fastest to within 5% since the modeling error is small and does not create much residual vibration.

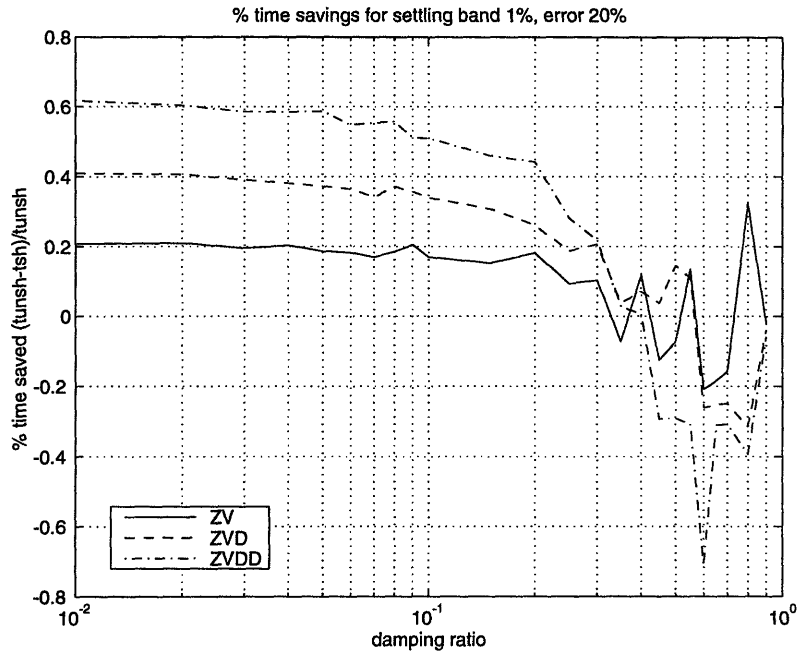


Figure 2.10: Time savings for a 1% settling band and a 20% error

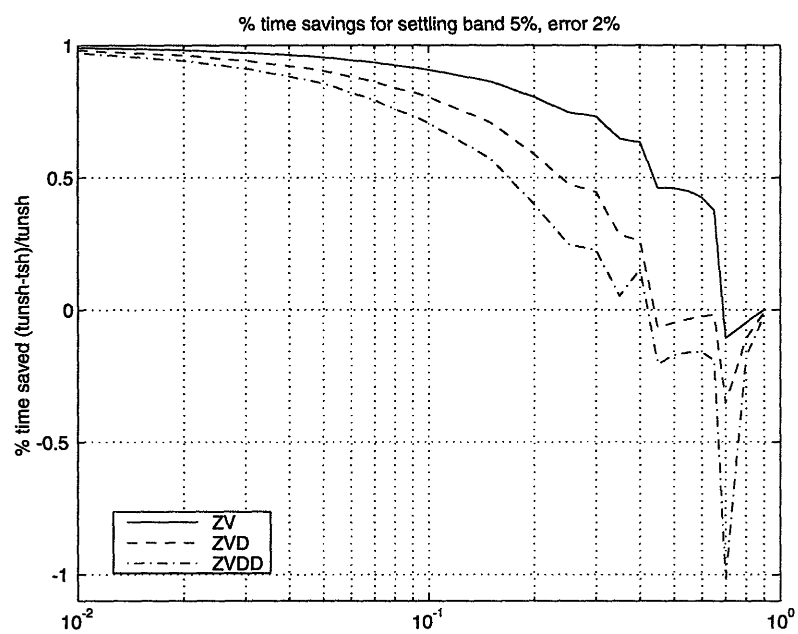


Figure 2.11: Time savings for a 5% settling band and a 2% error

Another interesting graph, shown in Figure 2.12, is for the ZVDD shaper for errors of 2, 5, 10, and 20% and a settling band of 1%. The 2, 5, and 10% error curves are all very close and there is no apparent difference in time savings between them. For the corresponding graph for the ZV shaper and 1% settling band, see Figure 2.13, where there is a larger difference between each error case. None of the curves save as much time as the ZVDD cases but they are all separated and parallel. If the amplitude band is changed to 5%, the ZV sequence saves a large amount of time for the 2% error case.

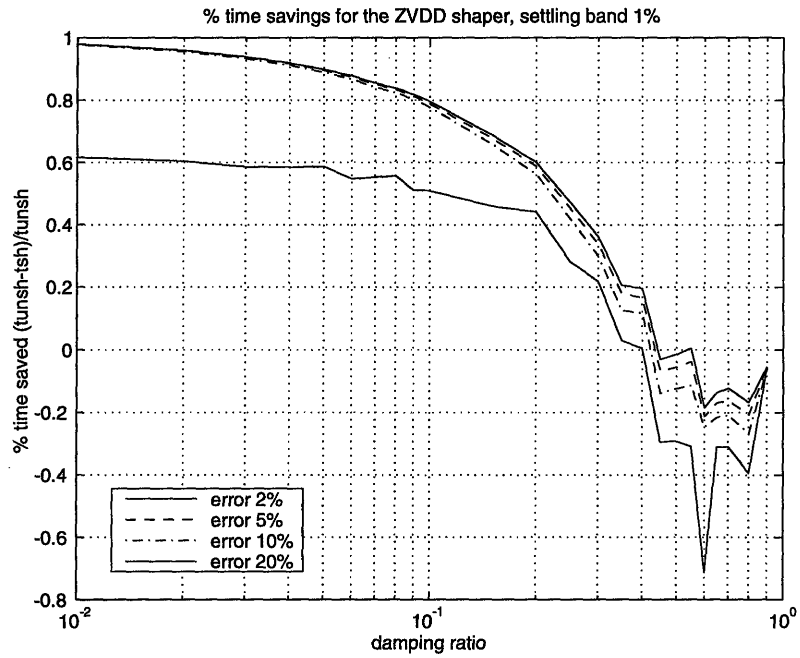


Figure 2.12: Time savings for ZVDD shaper

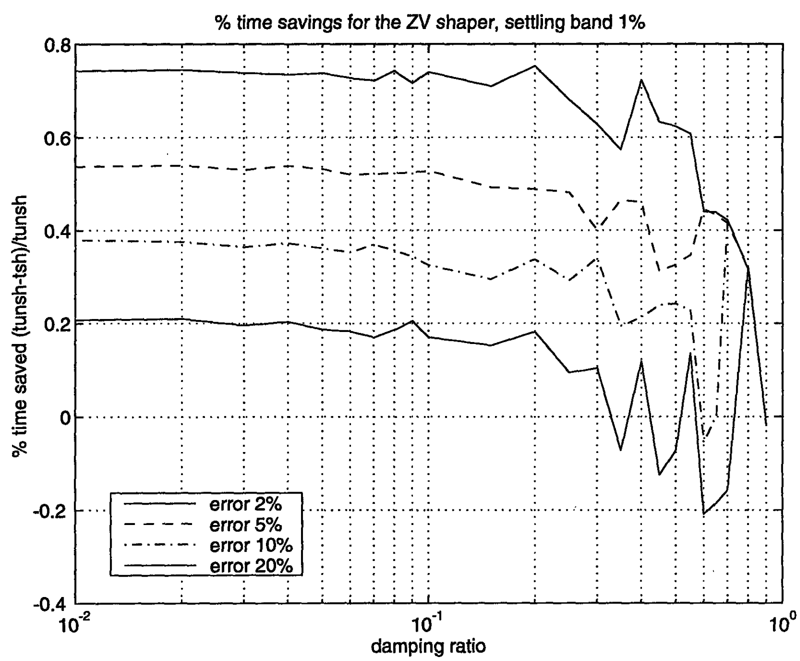


Figure 2.13: Time savings for ZV shaper

Test runs were done to see if changing the natural frequency mattered. Changing the natural frequency does change the settling time for the unshaped and shaped cases, but since % savings is a ratio, the ratio stays the same, independent of natural frequency. Trials were also done to compare the effects of raising the frequency shaped for instead of lowering it, see Equation 2.3. They do produce different results; increasing the frequency of the input shaper saves more time and is different from the decreased frequency case by about 4% for

e=0.02. The difference is more widely marked between the 5% error case. For a damping of .08 there is a 14% difference, while there is only a 4% difference between the two cases for a damping of 0.9. By using the graphs for the decreased frequency case, you are getting a more conservative estimate (less time saved), so the time saved will always be the same or higher than the estimate. The increased frequency case overestimates the savings, and so could lead to too high expectations of the possible time savings.

2.3.3 *Summary of Results*

Several tables of results of these simulations have been compiled. As shown in Table 2.3, for a small settling band of 1%, the ZVD or ZVDD shaper saves the most time for low damping ratios. As the damping ratios reach ζ=.2, the ZV shapers become the best choice. For the larger errors in knowledge, the ZVDD or ZVDDD shapers are the best choice. Interestingly, the unshaped response does the best when the damping is 0.9 or the error is above 10% and the damping is above 0.60. However, even at very high damping ratios, the unshaped has enough residual vibration that the shaped response still settles faster when the settling band is low.

	2% error		5% error		10 % error		20% error	
damping	best IS	% t saved	best IS	% t saved	best IS	% t saved	best IS	% t saved
0.01	ZVD	99	ZVD	99	ZVDD	98	ZVDDD	82
0.02	ZVD	97	ZVD	97	ZVDD	96	ZVDDD	80
0.03	ZVD	96	ZVD	96	ZVDD	93	ZVDDD	78
0.04	ZVD	95	ZVD	94	ZVDD	91	ZVDDD	76
0.05	ZVD	93	ZVD	93	ZVDD	89	ZVDDD	77
0.06	ZVD	92	ZVD	92	ZVDD	87	ZVDDD	73
0.07	ZVD	90	ZVD	90	ZVDD	84	ZVDDD	72
0.08	ZVD	89	ZVD	89	ZVDD	82	ZVDDD	70
0.09	ZVD	88	ZVD	88	ZVDD	80	ZVDDD	70
0.10	ZVD	86	ZVD	86	ZVDD	78	ZVDDD	66
0.15	ZVD	79	ZVD	79	ZVDD	66	ZVDDD	49
0.20	ZV	75	ZVD	73	ZVD	68	ZVDD	44
0.25	ZV	68	ZVD	63	ZVD	62	ZVDD	28
0.30	ZV	63	ZVD	55	ZVD	54	ZVDD	22
0.35	ZV	57	ZV	47	ZVD	42	ZVD	4
0.40	ZV	72	ZV	46	ZVD	40	ZV	12
0.45	ZV	63	ZV	31	ZV	24	ZVD	4
0.50	ZV	62	ZV	33	ZV	24	ZVD	15
0.55	ZV	61	ZV	35	ZV	23	ZV	13
0.60	ZV	44	ZV	45	none	0	none	0
0.65	ZV	44	ZV	43	none	0	none	0
0.70	ZV	42	ZV	42	ZV	42	none	0
0.80	ZV	32	ZV	32	ZV	32	ZV	32
0.90	none	0	none	0	none	0	none	0

Table 2.3: Results for a 1% settling band

Tables A.1 and A.2 were compiled for a settling band of 2% and 5%, respectively and are in Appendix A. One interesting result is that for the 2% error and 5% settling band, the ZV case is the best case for a range of dampings from 0.0 to 0.65. Only as the system near the critical damping does the unshaped response have the best settling time.

The main point is that as modeling error is increased, the faster and more sensitive input shapers cannot handle the error and settle quickly. A more robust shaper is needed; the wider sensitivity around the shaper frequency means that it reduces vibrations for a much wider range of frequencies. As the settling band is decreased, the same thing is true; more insensitive shapers are needed to reduce the residual vibration to even lower levels.

2.3.4 Conclusions

After looking at the results, it can be seen that there is a point at which input shaping is no longer useful. Once the damping ratio gets above 0.7, input shaping does not save much time because the unshaped response settles faster than the shaped responses, if there is a large error in system knowledge. This is especially true for the larger settling bands of 2% or 5%. As the modeling error gets larger, a more insensitive shaper is needed for the lower damping ratios. As the settling band increases, more residual vibration is allowed, so the less insensitive shapers perform the best. These tables should allow someone to pick an input shaper to use, if they know the approximate damping ratio of the system, the allowable amount of residual vibration, and the uncertainty of the system knowledge. For example, if the system damping is about 0.08, the settling band should be around 2% of the final value, and the error in system knowledge could be as high as 10%, then a ZVD input shaper should be used and the shaped settling time should be about 15% of the unshaped settling time.

Here are my recommendations about what input shaper to use and when it should be used:

- For a small settling band and a low error, use a less robust shaper such as ZV or ZVD.
- For a small settling band and a large error, use a more robust shaper such as the ZVDD if the damping ratio is less than 0.2. Otherwise use a ZV or ZVD shaper for higher dampings.
- For a large settling band and low error, use a ZV shaper. The fastest shaper is best here.
- For a large settling band and a large error, use a ZVD shaper for dampings below 0.2 and a ZV shaper for dampings above 0.2. If the damping is less than 0.05 and the error is greater than 15%, try an even more insensitive shaper such as a ZVDD or a EI shaper.

By appropriately using the various input shapers according to the sensitivity of system knowledge and the amount of residual vibration allowed, valuable time and control effort can be saved. The results of this study give novice users of the input shaping technology an idea of when and how to use input shaping and which shaper to implement.

2.4 Effects of Friction on Input Shaping

The purpose of this study was to determine if there was a connection between the bandwidth of a controller and the residual amplitude of vibration when input shaping is used. If a linear system is perfectly known, input shaping of the dominant modes will eliminate all of the residual vibration by the time the shaped command ends. For a system with a very low damping ratio, which implies that the system will "ring" for a long time in response to a step input, a input shaper is very effective at reducing the settling time. Usually a single mode ZVD input shaper gets rid of residual vibration within two cycles of vibration of the dominant mode.

It has been observed that when a higher bandwidth controller was used in conjunction with input shaping, there was less residual vibration after the move. [5] The system in question was a highly nonlinear flexible system, the MACE test article, with eight modes under 50 Hz. Three different bandwidth controllers, 3, 10, and 20 Hz, were given the same path to follow and input shaping was used to shape two modes with frequencies less than 10 Hz. The highest bandwidth controller had the highest percentage of vibration reduction and the absolute least residual vibration. This is despite the fact that the unshaped 20 Hz bandwidth slew caused much more vibration than the other unshaped slew for the lower bandwidth controllers. These observations were seen again when using a completely different system, which made this trend appear to be worth investigating. It was decided to develop a model of a multiple mode system and add nonlinearities to try to simulate the same effects.

2.4.1 System Description

First a linear multi-mode system was tested to see if the effects extended to the linear regime, though theoretically the effects should not. A two mass and spring system was chosen as the simplest example of a multiple mode system. The system is shown in Figure 2.14 and the equations of motion are given in equation A.1 in Appendix A.

The spring and masses were chosen to place the open-loop frequencies at 0 and 6 Hz. A force was applied to mass 1 and a proportional controller was added to close the loop. Thus the bandwidth of the controller could be increased simply by increasing the gain of the controller. This system has two modes, a rigid body mode and a flexible mode. Closing the loop by adding a proportional controller adds another vibratory mode to the system. For the rest of this section, the two modes referred to are the two vibratory modes, not the rigid body mode.

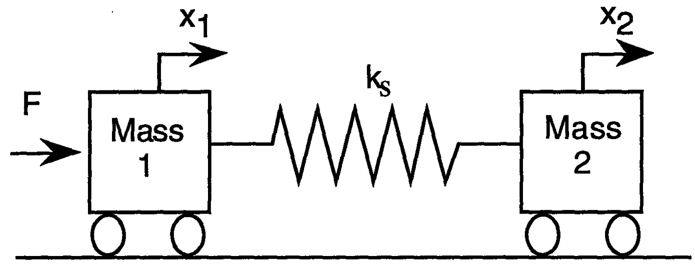


Figure 2.14: Two-mode system

Since the system was exactly known and linear, in theory shaping one or both of the vibratory modes should completely eliminate the vibration of the shaped for mode. This hypothesis was tested by running simulations using MATLAB. The results of the simulations showed that the input shaper did remove all residual vibration from the modes shaped for and therefore the bandwidth of the controller did not make a difference to the shaper. Simulations were done for the colocated (sensing and actuating the motion of mass 1) and the non-colocated (sensing the motion of mass 2 and actuating mass 1) cases. The results were the same for both the colocated and non-colocated cases; input shaping of a known linear system removed the mode completely.

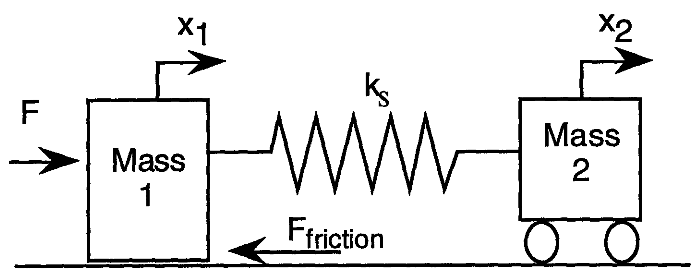


Figure 2.15: Nonlinear System

Next a nonlinear system was developed by adding friction to mass 1, as shown in Figure 2.15. A proportional controller was added to the system to close the loop and an input shaper was added before the loop. A simplified model of Coulomb friction was implemented where $F_{friction}=f_{mag}$ when the velocity of mass 1 was positive and $F_{friction}= -f_{mag}$ when the velocity of mass 1 was negative. The numerical integrator used was a MATLAB function called ode45 that integrates a system of ordinary differential equations using 4th and 5th order Runge-Kutta formulas and variable step sizes. Unfortunately, the step size was becoming too small because of the discontinuity in the friction model. A smoother model of friction was developed to deal with this problem and is shown in Figure 2.16. The relevant equations for this friction model are given in Appendix A.

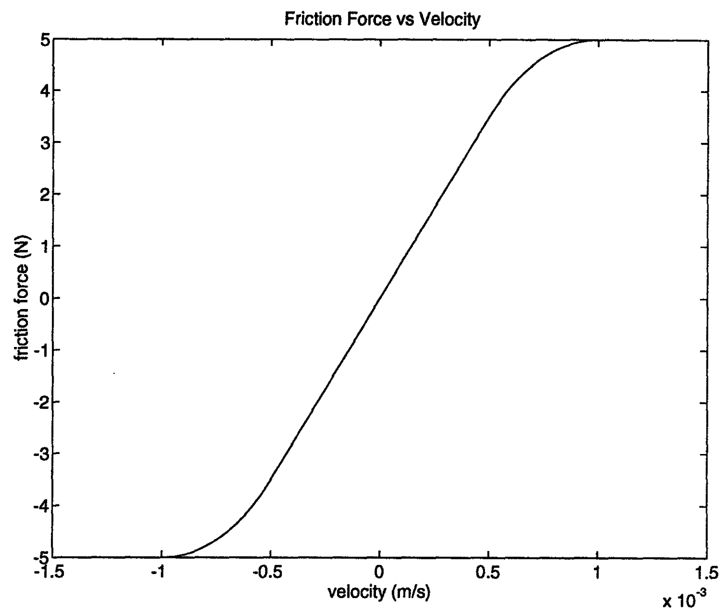


Figure 2.16: Smoothed representation of friction, f_{mag}=5 N

2.4.2 *Test Matrix*

Three different cases were chosen to be simulated. Case 1 was the non-colocated case with unequal masses, case 2 was the non-colocated case with equal masses, and case 3 was the colocated case with equal masses. In all cases the input force was applied to mass 1, thus colocated control implies actuating and sensing the position of mass one and non-colocated control implies actuating mass 1 and sensing the position of mass 2. Simulations were done for the following factors: different friction magnitudes, different proportional gains, insensitivity of the input shaper, and different combinations of modes to shape for. There were three different friction magnitudes: 2 N, 5 N, and 10 N. Then there were three different combinations of vibratory modes to shape for: shaping for low mode, shaping for high mode, and shaping for both modes. There were the unshaped case and three different kinds of shapers: ZV, ZVD, and ZVDD input shapers. The test matrix is given in Table 2.4.

Axis	Case 1	Case 2	Case 3
Friction (N)	2, 5, 10	2, 5, 10	2, 5, 10
Proportional gains	100-1000	100-900	100-1800
Input shaper	ZV, ZVD, ZVDD	ZV, ZVD, ZVDD	ZV, ZVD, ZVDD
Shaper frequency	ShL, ShH, ShB	ShL, ShH, ShB	ShL, ShH, ShB

Table 2.4: Friction test matrix

Each case also had a different number of proportional gains that spanned the stable space. Case 1 had 10 gains from 100 to 1000, case 2 had 9 gains from 100 to 900, and case 3 had 18 gains from 100 to 1800. The number of gains used in each case was dependent on the stability of the system. For cases 1 and 2, the system goes unstable at a gain of 1018 and 905 N/m respectively; these systems are shown in the first subplot of Figure 2.17. As can be seen from the lower subplot of Figure 2.17, the colocated system of case 3 never goes unstable. The smaller roots travel to zeros around 4.5 Hz while the larger roots travel to infinity. For cases 1 and 2, the frequencies of the modes travel toward each other as the gain is increased, while for the colocated case both of the mode frequencies increase as gain is increased.

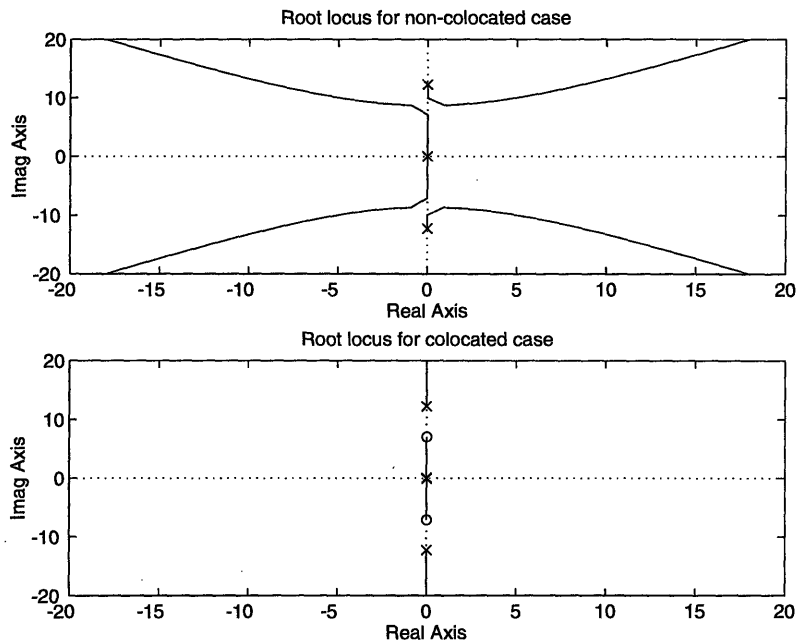


Figure 2.17: Root locus plots to demonstrate system stability

Another controlled variable was which combination of modes the shaper was designed for. Runs were done to shape for the low mode, high mode, and both modes. Normally one would not shape for the high mode alone, since the lower mode usually dominates, but it was included for the sake of completeness. These variables will be abbreviated as ShL, shaping for low mode, ShH, shaping for the high mode, ShB, shaping for both modes, and UnSh, no input shaping. The sensitivity of the shaper was another variable of interest. Originally the study only tested a ZVD shaper, but it was decided to see what the impact of the friction would have on the ZVDD shaper which is more robust, and on the ZV shaper, which is less robust. The ZVDD shaper is less sensitive to modeling errors, so it should reduce the residual vibration the most. The unshaped response, which could be called a one impulse shaped response, was found for every gain and friction magnitude. When shaping for only one mode, it is easy to calculate the shaper times and amplitudes using the mode frequency and damping. It is much more complicated to calculate the number of impulses, times, and amplitudes when shaping for two or more modes, since convolving two ZVD shapers does not necessarily give the time optimal solution. I used MATLAB functions created by Tuttle which calculate the optimal minimum time solution for an all positive input sequence. [32]

The friction magnitudes were chosen to span the largest possible range of values. In Figure 2.18, the unshaped and shaped for the high mode FFT amplitudes are plotted for a range of friction magnitudes. From the plot we see that F=2 N, 5 N, and 10 N span the space reasonably well. Friction magnitudes lower than 2 N approach the linear response, while magnitudes higher than 10 N approach the overdamped response where the two masses do not move. This study was done for a gain of 1000, which is in the middle of the colocated gain spread. For different proportional gains, the friction magnitudes span more or less of the response space, but K=1000 seemed to be a reasonable place to choose the friction magnitudes.

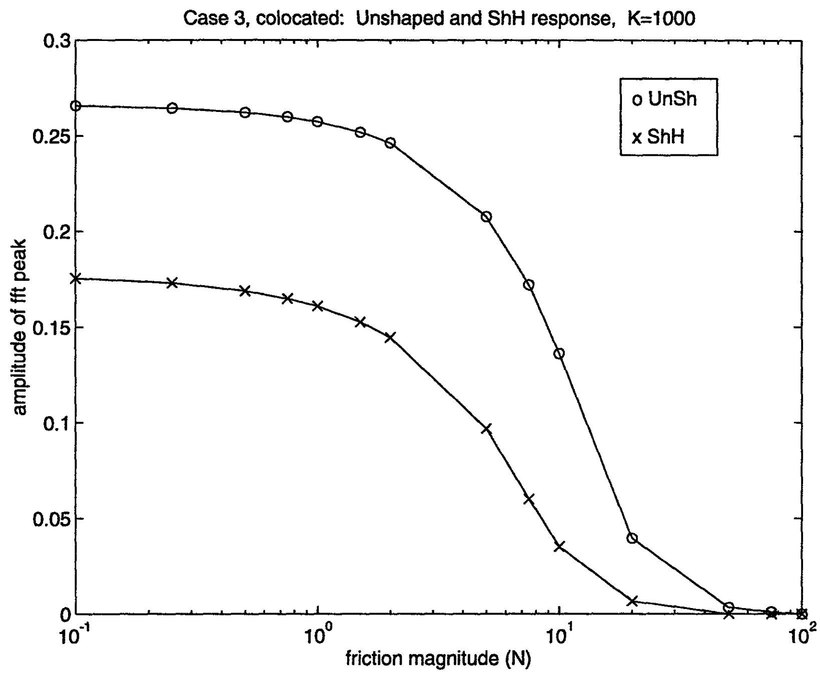


Figure 2.18: FFT amplitudes for case 3 as a function of f_{mag}

2.4.3 *Results*

Once the different variables were chosen, time histories were generated by MATLAB for the many cases and runs. A sample time history is shown in Figure 2.19 for a case 3 system run with a ZVD shaper and a friction magnitude of 5 N, which plots the displacement of mass 2. The four different lines represent four different types of shapers. The solid line is the unshaped response, the dashed is the shaped for low mode response, the dash-dot is the shaped for high mode response and the dotted is the shaped for both modes response. The low mode is dominating since the shaped for low mode and shaped for both modes responses are very similar, as are the unshaped and shaped for high mode time histories. The time delay associated with the input shaper is also visible. Since the time between impulses is proportional to the inverse of the frequency, the low mode shaper has a longer delay than the high mode shaper. But the ShL and ShB shapers get rid of most of the residual vibration within 0.6 seconds, while the unshaped system continues to ring for 10 seconds.

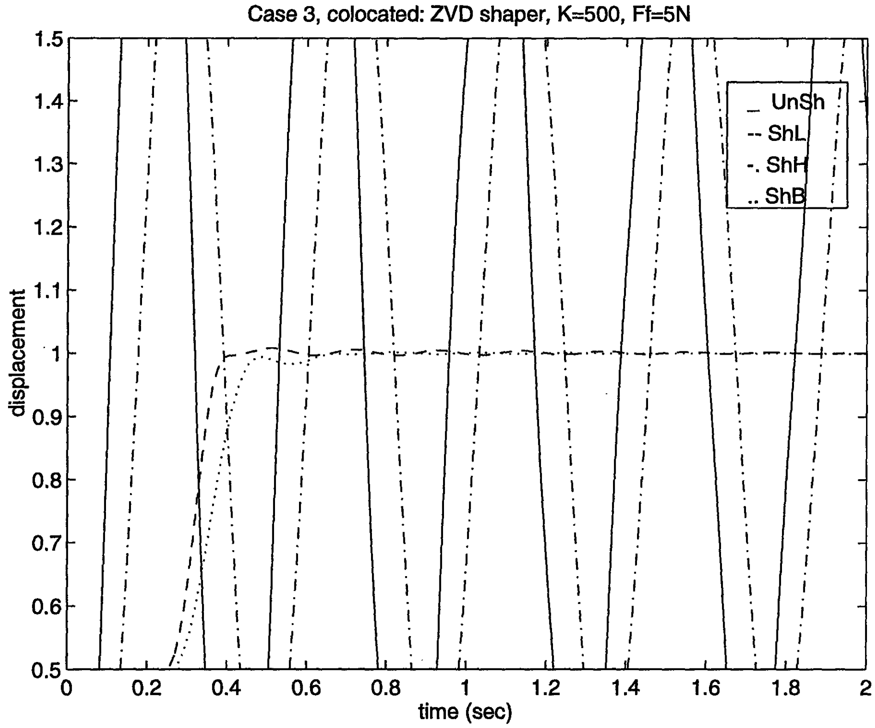


Figure 2.19: Time history of a case 3 step response

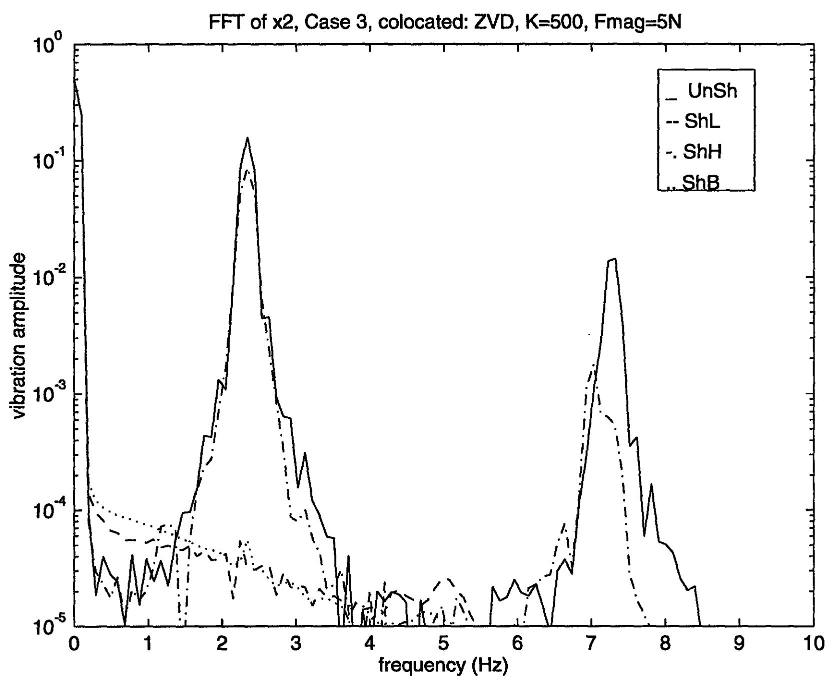


Figure 2.20: FFT of a sample case 3 time history

Once time histories were generated, a way of evaluating them was needed. In a single mode system, it is fairly easy to find good performance metrics such as settling time and residual vibration after shaped move is over. However, finding a valid performance metric is not as simple for a multiple mode system. The amount of residual vibration cannot be pulled off a time history since the modes add together and overlap. So the data was taken to the frequency domain where a Fast Fourier Transform was taken of the windowed data. A Hanning window was used, since the data was not periodic. Figure 2.20 shows the windowed FFT of the time history shown in Figure 2.19. Here the large difference in residual vibration can be quantitatively measured. The unshaped low mode peak is at least three orders of magnitudes larger than the ShL peak and the ShB peak.

Once in the frequency domain, a subroutine was written and implemented in MATLAB to find the modes and the corresponding amplitudes, as well as to sort the modes by amplitude and frequency. In order to differentiate between actual mode peaks and noise in the data, the lower limit of the FFT vibration amplitude was set to be 0.001. This meant that if the vibration was occurring in a band of less than 0.001 around the final position of 1.0, the subroutine assigned the residual vibration an amplitude of 0.0. This was a logical step, because at this amplitude level the oscillations are so small that they do not affect the settling time. This also cuts out any noise in the frequency, which seems to be below 10^{-4} in Figure 2.20. This subroutine generated many tables of data which had to be analyzed.

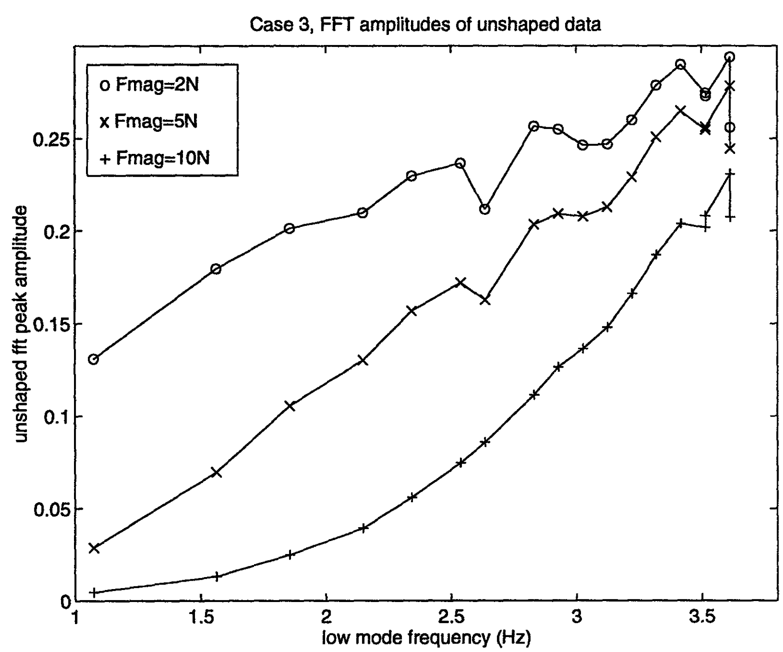


Figure 2.21: FFT amplitudes for case 3, unshaped amplitude vs. low mode frequency

The easiest way to find trends in data is to look at it graphically. Since there were so many varied parameters, there were many different ways to display the data graphically. I decided to plot the amplitude of the FFT peak vs. the frequency of the low or high mode, with three lines of varying friction magnitudes on each graph. (This is almost the same thing as plotting amplitude vs. controller gain since the gain and the frequency of the low mode increase together in all cases.) Figure 2.21 shows this configuration for the unshaped data of case 3 for the low mode frequency. The lines start out far apart, but as the gain increases, the lines draw together. This can be explained by the fact that the friction effect becomes less important as the gain overwhelms it. Thus, at high controller gains, the system could be modeled as a linear system where friction is ignored.

The FFT amplitudes were plotted for every run in every case. Figure 2.22 and Figure 2.23 show the unshaped amplitudes for all cases and both the low mode and high mode frequencies. Note that in Figure 2.23, the scale is different for the case 3 plot. In Figure 2.27, the amplitudes are plotted versus the low mode frequency, all of the amplitudes are increasing with gain. The non-colocated amplitudes are increasing much more and to higher amplitudes than the colocated ones. As the non-colocated systems approach instability, the systems ring more and thus the FFT amplitude is higher since the modes aren't damped. The difference in total amplitude is quite large, since case 3 gets up to 0.3 while cases 1 and 2 get up to 1.45 (in reference to the input step of magnitude 1.0). As the controller gain increases, the high mode frequency decreases for the non-colocated cases and increases for the colocated case. This means that as the proportional gain in case 1 or 2 is increased, the high mode frequency actually is decreasing and moving from high to low frequency. Case 3 is exactly the opposite; as the proportional gain is increased, the high mode frequency is also increasing and moving from low to high. These amplitudes are less than the first mode amplitudes, much less in the colocated case which is lower by an order of magnitude. Another observation is that as the friction magnitude increases, the amplitude decreases. Figure 2.22 and Figure 2.23 show the trends in the unscaled data.

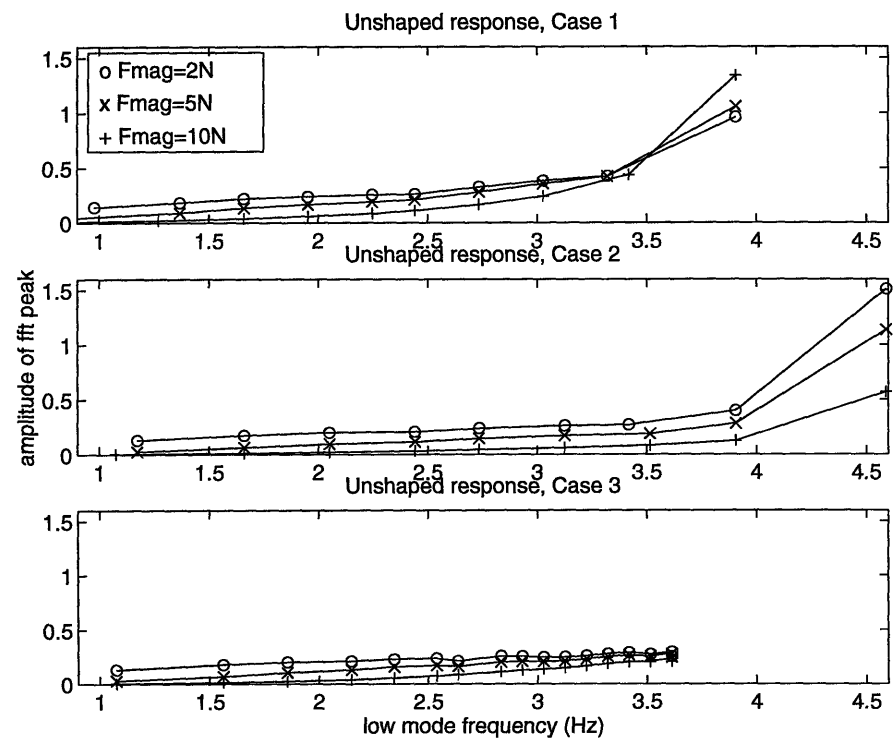


Figure 2.22: Unshaped low mode amplitudes

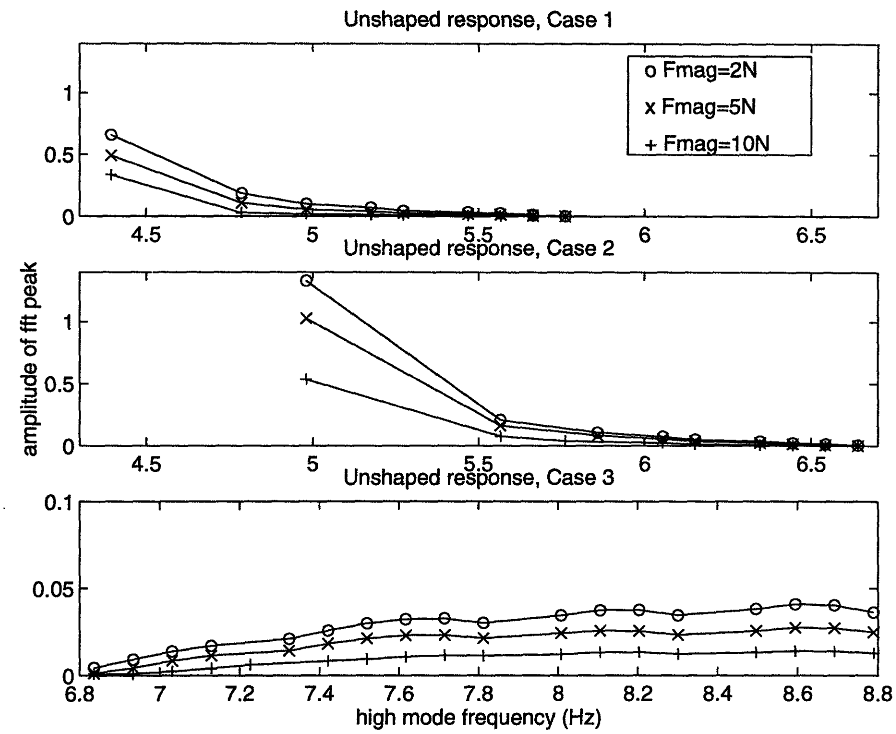


Figure 2.23: Unshaped high mode amplitudes

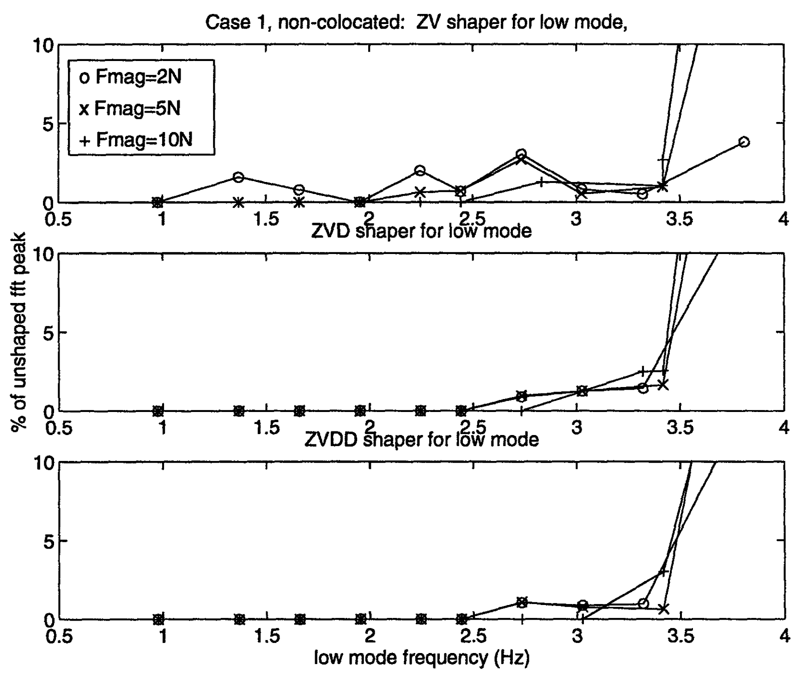


Figure 2.24: Case 1 scaled low mode amplitudes when ShL

Unfortunately, it was hard to compare between different runs and cases without a baseline reference. To make this task easier, the data was scaled by dividing each data point by its unshaped counterpart, which gave the percentage of vibration reduction. Figure 2.24, Figure 2.25, and Figure 2.26 show the percentage of the unshaped vibration left after the move finished plotted versus the low mode frequency.

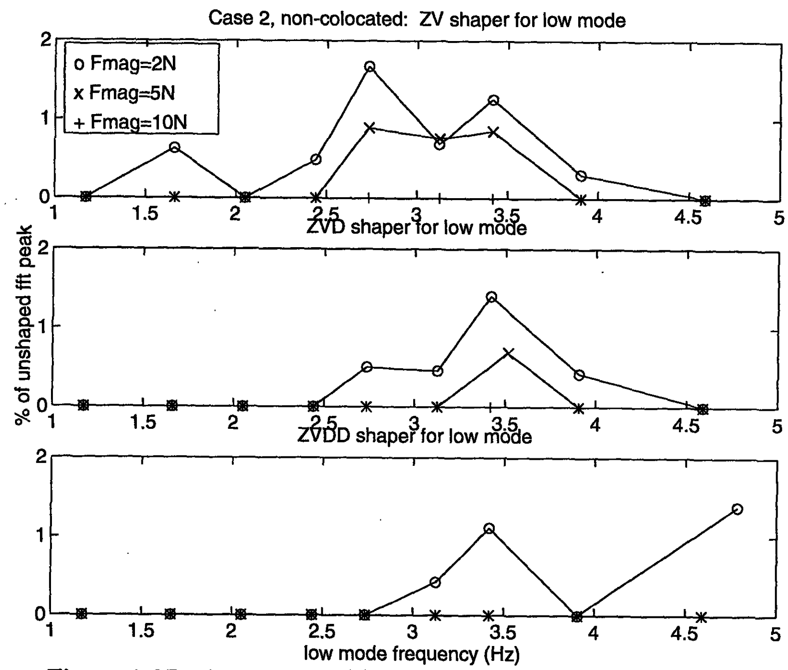


Figure 2.25: Case 2 scaled low mode amplitudes when ShL

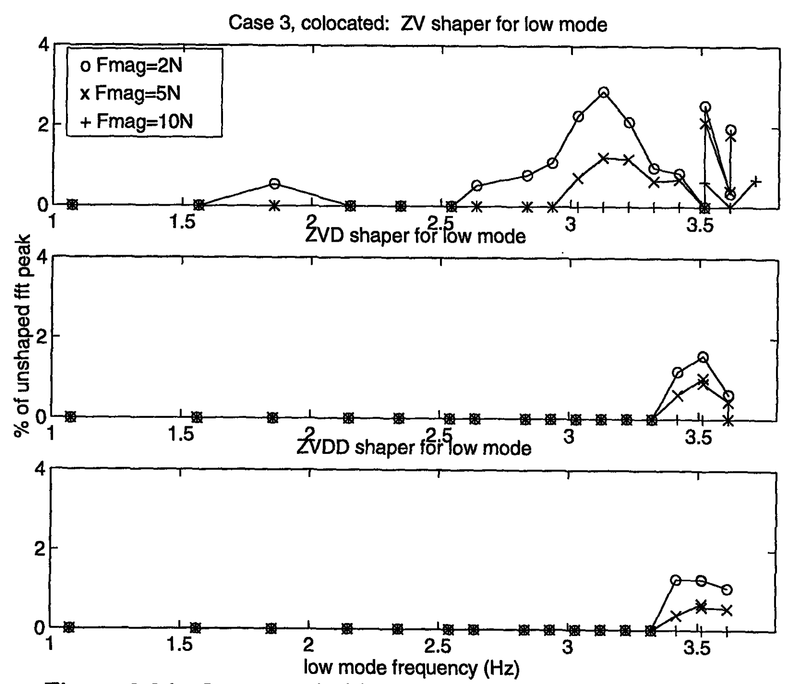


Figure 2.26: Case 3 scaled low mode amplitudes when ShL

The above figures are only a small portion of the total number of graphs. These figures are the most interesting and show the shaping for the low mode cases, low mode frequencies and amplitudes only. Similar graphs could be produced for the high mode when ShL, but no broad conclusions could be drawn. When looking at the high mode frequency and the shaping for the low mode case, the percentage of the unshaped high mode amplitude does actually decrease below 100%. This is partly because input shaping takes out energy from the command, and thus excites less vibration. But the low mode itself plays a part, since the input shaper takes out the low mode and three times the low mode and five times the low mode, etc. For all cases, if the high mode frequency, f_2, and the first few harmonics of the low mode, f_1, are plotted, as in Figure 2.27, the lines cross, indicating mode cancellation at that frequency. The high mode amplitude for ShL is actually higher on average for the non-colocated cases, because the lines don't cross for as long, while in the colocated case the 3^*f_1 line runs almost parallel to the f_2 line.

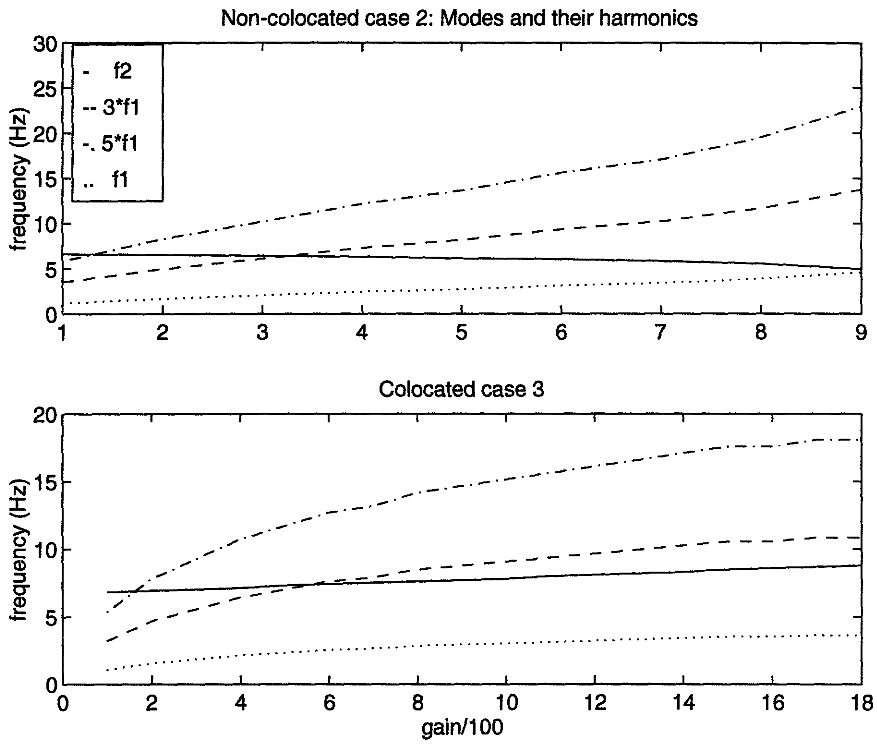


Figure 2.27: Modes and harmonics

In all of the shaping for the low mode cases, the input shaper reduced the first mode's residual vibration amplitude to under 4% of the unshaped amplitude, except near the instability in the non-colocated cases. For the ZVD shaper, the results are all below 2%, except for the last two gains in case 1. The ZVDD shaper cases do reduce the residual vibration by more than the ZVD cases, but not by enough to justify the time delay that the ZVDD shaper produces. The ZV shaper has good results when the friction magnitude is equal to 10 N, but otherwise leaves considerably more residual vibration than the more robust shapers. There is no particular pattern in the data, only a tendency for the low mode amplitudes to be lower at lower gains, and increase some as the gain increases. This is because the unshaped low mode amplitude is increasing with the controller gain. Thus the input shaper has to take out more total vibration to get the same % of unshaped amplitude. For example, in case 3 with the friction magnitude equal to 5 N, at K=200 the unshaped low mode amplitude is 0.0695, while at K=1500 the unshaped low mode amplitude is 0.2546. This does not make a difference to the input shaper in linear systems, but the friction seems to affect the results here. The friction adds damping to the system and reduces residual vibration, but it is also delaying the system response.

In the shaping for the high mode cases, which are not shown, the unshaped amplitude was much lower for the high mode than the low mode. The low mode amplitude's maximum occurs at 0.3, while the high mode's maximum only goes up to 0.05. Therefore, it was easier to lower the second mode's residual vibration below the limiting threshold of 0.001. The performance of the input shapers was much better for ShH and got rid of the vibration almost completely for the ZVD and ZVDD shapers. The ZV shaper did not work as well and left a few runs with a low amount of residual vibration. In the ShH cases, the low mode amplitude was close to the unshaped value, unlike in the ShL cases where the low mode shaping affected the high mode. When ShH, the low mode amplitude does decrease for the non-colocated cases since as the high mode approaches the low mode, the input shaper starts to reduce the low mode amplitude. However, approaching instability also cause the peaks at the nearly-unstable gain to increase dramatically, as shown by the low mode ShL amplitudes in Figure 2.24. The ShH cases are not as useful, since you do not usually shape for the higher mode alone. Here the friction does not really affect the shaping as much as approaching instability does in the non-colocated case. In the colocated case, the shaper gets rid of the high mode at least 16 out of 18 times for all friction magnitudes and number of impulses in the shaper.

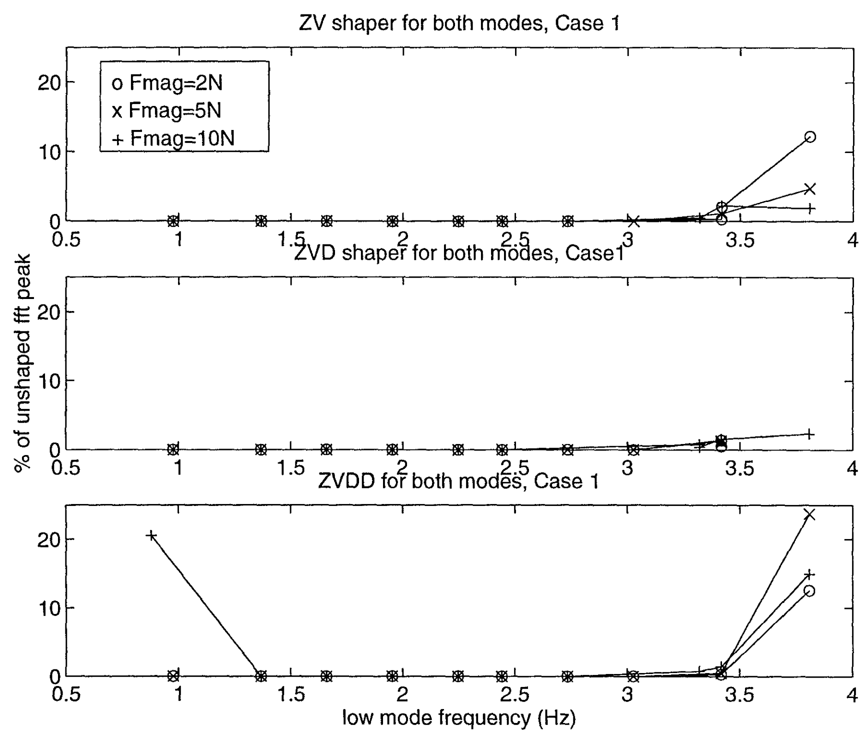


Figure 2.28: Scaled low mode amplitudes when ShB, case 1

All of the colocated runs had zero residual vibration when shaping for both modes. The non-colocated case 1 did have some residual vibration in the first mode at higher gains as instability was approached, as shown in Figure 2.28. Case 2 had some interesting results. In this equal mass, non-colocated case, at higher gains and friction magnitudes another mode appeared at 4.78 Hz. This corresponds to the motion of mass 2 and the spring, while mass 1 is motionless. The mode frequency is given in Equation 2.5. This mode showed up at gains

$$f = \frac{1}{2\pi}\sqrt{\frac{k_s}{m_2}} \tag{2.5}$$

above 500 N/m in the f_{mag}=5 and 10 N runs for ShL, ShH, and ShB. This indicates that the friction was stopping mass 1 and allowing mass 2 to vibrate freely. Since there was no friction on mass 2, in the simulation it could ring forever, which is not very realistic. The mode at 4.78 dominated the higher mode, which did not appear when the mass 2 mode appeared. There was no way to reflect this trend in the summary tables, so case 2 was left out and only cases 1 and 3 are summarized and in Table 2.5 and Table 2.6.

Case 1	Number of zero amplitudes in low mode (maximum is 10)			
	shaper	F=2N	F=5N	F=10N
Shaping for	ZV	2	4	6
Low Mode	ZVD	6	6	7
	ZVDD	6	6	8

Case 1	Number of zero amplitudes in high mode (maximum is 10)			
	shaper	F=2N	F=5N	F=10N
Shaping for	ZV	8	9	9
High Mode	ZVD	10	9	10
	ZVDD	10	10	10

Case 1	Number of zero amplitudes in low and high modes (maximum is 10/10)			
	shaper	F=2N	F=5N	F=10N
Shaping for	ZV	7/10	8/10	7/10
Both Modes	ZVD	8/10	8/10	6/10
	ZVDD	8/10	8/10	6/10

* zero amplitude implies residual amplitude less than 0.001

Table 2.5: Summary table for case 1

The numbers in the cells represent the number of FFT modal amplitudes which were below the lower threshold (of 0.001) in a certain set of runs. These numbers are a measure of how well the input shaper removed the residual vibration. The higher the numbers, the less residual vibration there was in general in each set of runs. Each cell represents a set of tests for a certain number of impulses and a certain friction magnitude run over the applicable range of controller gains. The number of zero amplitudes is summed for the set of controller gains and placed in the table. For the ShB modes runs, the first number is for the first mode and the second number represents the second mode. The tables show that the input shaper does better, in general, as the number of impulses or the friction magnitude increases. There does not appear to be too much difference between the ZVD and ZVDD shapers, so it makes sense to go with the ZVD shaper which gives less time delay.

Case 3	Number of zero amplitudes in low mode (maximum is 18)			
	shaper	F=2N	F=5N	F=10N
Shaping for	ZV	6	10	16
Low Mode	ZVD	13	14	18
	ZVDD	13	13	18

Case 3	Number of zero amplitudes in high mode (maximum is 18)			
	shaper	F=2N	F=5N	F=10N
Shaping for	ZV	16	18	18
High Mode	ZVD	17	18	18
	ZVDD	18	18	18

Case 3	Number of zero amplitudes in low and high mode (maximum is 18/18)			
	shaper	F=2N	F=5N	F=10N
Shaping for	ZV	18/18	18/18	18/18
Both Modes	ZVD	18/18	18/18	18/18
	ZVDD	18/18	18/18	18/18

* zero amplitude implies residual amplitude less than 0.001

Table 2.6: Summary table for case 3

The runs with the most variations between runs are the shaping for the low mode simulations. For both case 1 and case 3, the shaper for both modes gets rid of all or most of the residual vibration. The shaper for the high modes does the next best job of reducing vibration, mostly because there is not as much vibration to get rid of in the first place, since the second mode does not dominate. The reason why some of the trends seem wrong is that there were many points right near the arbitrary threshold value of 0.001. Under this system 0.0009 was set equal to zero, while 0.0011 stayed the same. This caused the variation in results in the shaping for both modes and case 1 runs. The cells containing 6/10 actually could be called 8/10 if the threshold was changed to 0.002, for example. The colocated input shapers seem to remove vibration more effectively than the non-colocated shapers. This is a function of the system, since the actuator force is applied directly to the colocated mass, but is applied through the spring to the non-colocated mass. There is no spring in its way to cause a time lag or reduce the effectiveness of the input shaper.

2.4.4 Conclusions

A simple nonlinearity does reduce the effectiveness of shaping. However, the input shaper still reduces the vibration amplitude to a maximum of 3% of the unshaped amplitude for the colocated case. In the non-colocated case, stability issues cloud the results, but the vibration amplitude is still reduced to at least 5% of the unshaped amplitude in all cases except for the highest gain which is very close to instability. If there is a requirement of reducing the residual vibration to less than 5% of the original unshaped vibration, here are some guidelines. A multiple mode shaper will always reduce the vibration to below 5% in a colocated case, so a multiple mode shaper with the insensitivity of a ZV shaper will work fine. However, in the non-colocated case, you should use a multiple mode ZV shaper unless the system is very near instability, in which case a multiple mode ZVD shaper should be used. If shaping for the dominant first mode only, avoid controller gains near instability, since the unshaped high mode will excite too much vibration. Use a ZV shaper to meet the 5% band, while a ZVD shaper will meet the 2% vibration reduction criterion. If shaping for a second mode only, use a ZV shaper if there is friction, or a ZVD shaper if there is very little or no friction. In general, friction adds damping to the system and helps reduce the system vibration, so a lower input shaper can be used with higher friction. Just beware of the lack of robustness associated with a ZV shaper. If the system frequencies are not precisely known, use a more robust ZVD shaper.

Here are some recommendations:

- If the residual vibration must be reduced to a very small percentage of the unshaped, use a multiple mode shaper (ZV/ZV).
- If you want to use a single-mode shaper, shape for the lower mode.
  For reduction to 5% of unshaped, choose ZV.
  For reduction to 2% of the unshaped, choose ZVD.
- If the system is near instability, use a more robust ZVD shaper and shape for both modes.
- With more friction, a faster shaper (ZV) can be used to take advantage of the increased system damping.

I did not see the higher bandwidth controllers reducing the residual vibration at all. In fact, the results seemed to indicate the opposite in that at higher controller gains the system has less percent vibration reduction. This can be explained by the fact that at higher bandwidths more energy is sent to the system and thus more vibration is excited, which leads to more vibration for the input shaper to get rid of.

This study showed that a simple nonlinearity like friction does not impair the input shaper's performance much at all. Even with varying friction magnitudes it was possible for a single mode shaper to obtain a 96% vibration reduction compared to the closed loop, proportional controller step response. The multi-mode shapers performed even better and reduced the vibration to well below the limiting threshold of 0.001 except near instability points.

The FLEX Program

Chapter 3

3.1 Introduction

Input shaping has been applied to many systems since it was first developed by Singer. [25] Space is particularly well suited for the use of input shaping because of the restrictions on the weight of components. The dynamic characteristics of the large, lightweight flexible space structures are very different than those of industrial robots. The payload-to-arm mass ratio can be on the order of 100:1 for a space manipulator whereas an industrial arm has a ratio of less than 1. [22] The light and flexible space structures are also very lightly damped with low frequency modes, so moving can cause a lot of residual vibration. These residual vibrations then cause large time delays. Input shaping does have an inherent time delay, but compared to the duration of vibrations for a very lightly damped structure, input shaping saves a lot of time by eliminating vibrations.

The FLEX program is one of the NASA In-Step programs. Its main goal is to define the best methods for controlling construction, inspection, and repair systems in space. The FLEX team consists of MIT, Martin Marietta, Payload Systems, Inc., and Convolve, Inc. The principal investigator is Professor Warren Seering of MIT. Other members of the MIT team were Dave Miller, Ketao Liu, Carl Blaurock, and Bill Singhose. Ketao Liu created the FLEX models and Carl Blaurock designed all of the feedback controllers. This chapter will discuss the different tasks accomplished during Phase A of the FLEX program, including modeling, designing feedback controllers, and investigating feedforward methods.

3.2 Project Motivation

Current Shuttle Remote Manipulator System (SRMS) operations are severely constrained by performance limitations. A study by Newsome, et. al, found that approximately 30% of the operational time for the SRMS is spent waiting for vibrations to decay to within acceptable levels. [17] As heavier payloads such as International Space Station Alpha (ISSA) components are grappled by the arm, oscillation periods will become even longer. Arm vibrations are a serious problem for astronauts for three operating conditions: quick movement of the unloaded arm, manipulation of payloads with flexible components, and precise positioning maneuvers.

Increased settling time is not the only penalty associated with arm oscillation. Precise motions are very difficult to perform with a robot that tends to oscillate. The lower the vibration frequency, the more difficult the precision task. Oscillations are likely to create problems for astronauts trying to orient and precisely locate large payloads during construction of the ISSA. Residual vibration will be an especially serious problem when the SRMS and comparable arms are used to handle payloads of masses near their load limits.

State-of-the-art teleoperation control methods are not currently in use on the SRMS. They are not planned for use on arms being developed because these technologies have not been evaluated sufficiently under realistic operating conditions to fully explore their effectiveness. Unfortunately, testing these technologies on the SRMS is prohibitively expensive; the cost, time required, and risk associated with modifying the SRMS software make such tests impractical. The FLEX program has been designed as a practical and cost effective alternative to a test program employing the SRMS.

3.3 Objective

The overall objective of FLEX is to conduct a set of experiments which will define the best methods for controlling construction, inspection, and repair systems in space. We must also establish the value of state-of-the-art vibration and teleoperation control methods for improving the performance of construction, inspection and repair systems in space.

There were several specific objectives accomplished during Phase A. First, members of the FLEX team have developed a fully three dimensional model of a manipulator representative of those to be used for construction and maintenance tasks for ISSA and other future space systems. Such a model is a key component in the development of effective control methods. For this phase we chose to model the current SRMS because it represents the state-of-the-art in manipulators for use in space and because data necessary for validating such a model was available from JSC and SPAR Aerospace. Second, we evaluated the performance of an array of candidate control methods and, through extensive testing in simulation, documented the potential of the most promising methods for improving the performance of large, flexible robots. Feedback control methods were researched, tested, downselected, extensively tested and downselected again. Feedforward methods were reviewed and selected, then implemented on the SRMS simulation with the feedback controllers. Comparisons were made between simulations with and without feedforward control. Third, we conducted ground tests to determine whether providing information about the states of a manipulator to the astronaut operating the manipulator had the potential to improve task performance. Fourth, we considered a number of candidate robot designs for an on-orbit test facility and selected one that will allow us to achieve all of our technical objectives safely and at a reasonable cost. The remainder of this chapter details our work on achieving the first two of these goals. For more detailed information on the FLEX Phase A work, see reference [23].

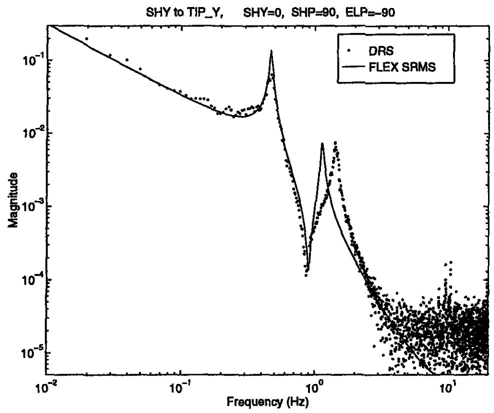
3.4 Modeling

During Phase A, we generated a symbolic, recursive, order N, Lagrangian formulation of the equations that describe the motion of a fully three-dimensional arm model. The model platform was designed so that with inclusion of appropriate system parameters, it can represent a wide array of flexible manipulators including the SRMS, the Space Station RMS, the test manipulator that has been proposed for our on-orbit studies, and the ground based test system that we plan to use in preparing for the proposed mission. The availability of the symbolic equations is important for the model parameter derivation procedure, which will use ground test data describing individual components where possible, along with sensitivity relations derived from the symbolic equations, to refine the nonlinear model.

Using this model platform, we created and calibrated a simulation of the SRMS, called the FLEX simulation, to serve as a test bed for our control method comparison studies. An accurate model of a three-dimensional manipulator in space is crucial to stable implementation of advanced controllers on orbit, and consequently to the success of the program. Also, by evaluating the controllers on a model of the SRMS, we have illustrated the impact that these controllers could have if transferred to the SRMS.

The MATHEMATICA program was used to derive nonlinear and linearized symbolic equations of motion of the SRMS. These symbolic equations were then transferred to FORTRAN code for the nonlinear model and MATLAB code for the linear model by a C program. Because the symbolic equations were readily available, we could easily obtain sensitivity equations of the nonlinear model for model updating using experimental data. The availability of the equations also let us have direct control of the modeling accuracy and nonlinear characteristics of the models.

The model was designed to represent the modal characteristics of the SRMS up to 20 Hz. The models were derived using the assumed mode method. Joint and base flexibilities were included in the shoulder joint, elbow joint, and the base models. Flexibility in the links was also included in the model. Parameter values for system components were obtained from reports generated at JSC. We calibrated our model by comparing its performance with that of the Draper Remote Manipulator Simulation (DRS) which has been verified against flight data and is considered to be a very high fidelity model. The model effectively represents the dynamic performance of the SRMS, particularly for the dominant first mode. For more information, see reference [23]. Figure 3.1 presents the frequency responses (FRs) from joint velocity commands to tip position for our FLEX model and the DRS. As the figures show, the model effectively represents the dynamic performance of the SRMS, particularly for the dominant first mode.



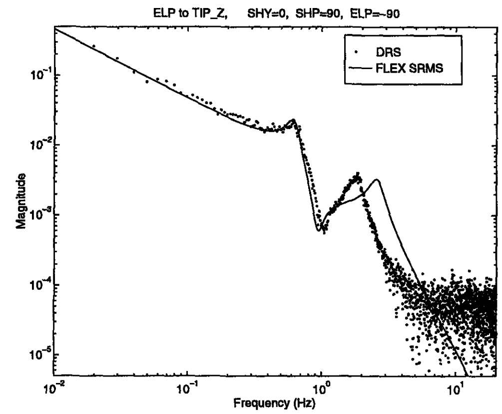


Figure 3.1: FR from shoulder yaw to tip y and FR from elbow pitch to tip z

Several different versions of the SRMS simulation were created during the program. The first model, $FLEX_1$, was a planar model with no payload and very little Coulomb friction. Its inputs were the desired shoulder pitch and the elbow pitch torques. The second model developed, $FLEX_2$, was a planar model which allowed variable payload masses and included nonlinear Coulomb friction effects. The final model of the SRMS is a fully three-dimensional model, $FLEX_3$. It has three rotary joints with two flexible links and a variable payload. The first joint is the shoulder yaw joint with a range of motion of ±180 degrees. The second joint is the shoulder pitch joint, which can move from -2 degrees to 145 degrees. The third joint is the elbow pitch joint, which is located between the first and second link. Its workspace is from 2 degrees to -160 degrees. The FLEX model's workspace matches that of the SRMS. The first link is 20 feet long, 21 when the joints are included, and the second link is 23 feet long, 28.2 when the joints and wrist are included. The flexibility of the FLEX simulation is 50% in the joints, as opposed to the usual 90% in the joints for commercial machines.

The various models of the SRMS developed during the FLEX program will be called the FLEX simulations in the rest of the thesis, to distinguish them from other models used.

3.5 Feedback Controllers

Most existing control methods fall into one of three categories: linear control, nonlinear extensions of linear control, and true nonlinear approaches. Proposed linear methods include proportional plus integral plus derivative (PID) independent joint control, PD plus constant gain link strain feedback (PD + strain or PDS), and linear dynamic compensation such as the Linear Quadratic Gaussian (LQG) controller. Nonlinear extensions include gain-scheduled LQG, H-infinity control, and Linear Quadratic Regulator (LQR) and LQG methods with nonlinear state estimators. These methods are well understood and account for geometric nonlinearities of the SRMS. Nonlinear techniques encompass adaptive control such as Recursive Least Squares (RLS) and variants, as well as hybrid designs which combine nonlinear joint and angle control with linear control of flexibility. Work to date suggests that nonlinear control alone tends to give poor performance for flexible systems, possibly since theories for providing accurate state measurement or estimation are not sufficiently well understood.

The feedback control investigation is being conducted on the SRMS so that many of the control issues that are common to flexible manipulators (joint flexibility, nonlinear friction, etc.) will be investigated. Additionally, examining the behavior of the Shuttle manipulator under feedback control will highlight the control problems inherent in that configuration which result from high gear ratios, extremely high payload-to-arm ratios, and so on.

We began the feedback analysis with a preliminary evaluation of the many feedback controllers which have been proposed for the SRMS. We want to understand the value of state-of-the-art software for existing arms and for the next generation of arms which are already designed. Therefore, the best controller for the SRMS can be selected by discarding those which do not control flexibility, and those which would require substantial SRMS hardware modification (such as the addition of link piezoelectrics as actuators, or of tip position sensing). The remaining controllers were ranked based on formulation complexity (tractability of the derivation in three dimensions), implementation complexity (computational requirements and number and type of sensors used), and maturity (ground test heritage).

The initial downselect was based upon a literature review and the above criteria. The surviving controllers were the PD controller, the sensitivity-weighted LQR, multiple-model LQR, feedback linearized plus PD, gain-scheduled LQR, and feedback linearized plus multiple LQR. These feedback controllers were designed for the first planar model, $FLEX_1$, and its particular configuration space of no payload and very little Coulomb friction.

The proportional-derivative (PD) independent-joint feedback controller is the baseline control to which all other controllers will be referenced. As predicted by other researchers, the closed-loop bandwidth is limited by the first zero frequency. Additionally, the architecture limits the closed-loop damping of the flexible modes, leading to substantial oscillations at the tip, even if the joint angles are well-controlled. To achieve good positioning of the end-effector, the arm controller must account for flexibility in the link.

One technique which accounts for joint and link flexibility is the Linear Quadratic Regulator (LQR) theory. To examine the LQR on the SRMS simulation, the arm was linearized about a 90 degree elbow configuration. The optimal feedback matrix was found by penalizing oscillations of the tip. The

LQR feedback did not control "pinned-pinned" modes (which do not control displacements of the tip), leaving them lightly damped. A penalty on the encoder outputs was incorporated, which resulted in closed loop damping in all modes. LQR feedback was not stable for the entire range of motion of the arm. A technique called sensitivity weighting was used to uniformly add damping to all closed loop modes. This resulted in stable operation throughout the workspace for the SWLQR. It was optimized around θ_{elb}=90 degrees. Desired velocities can be fed to the SWLQR-controlled system, enabling velocity rather than position regulation. The SWLQR bandwidth (0.6 Hz) is approximately that of the PD (0.55 Hz), but the settling time is roughly twice as fast. The improvement in tip positioning is solely due to the inclusion of flexible effects in the controller.

The gain-scheduling LQR (GSLQR) controller was implemented by partitioning the workspace into regions (within which the arm response is nearly linear). Then a LQR gain matrix was determined for each region. Multiple model LQR design was developed by forcing a single gain matrix to control the arm at several locations in the workspace. The MMLQR has equal weighting on settling times around the elbow angles of 90 degrees and 10 degrees. SWLQG, MMLQG, and FBLQG controllers were also designed for the initial FLEX model. The decision was made to design LQG controllers because a large number of sensors would be needed for a LQR controller. There are a limited number of sensors available on orbit, and the LQR needed more sensor inputs than were available. More sensors could be added to the SRMS or the FLEX test arm, but it would be too expensive.

The feedback linearizing controller was designed with both a PD controller outer loop (FBL) and a MMLQR controller outer loop (FBLMM). It combines the rigid robot technique of inverting the rigid body dynamics and an outer loop closed around the feedback linearization. With a flexible arm, the rigid feedback linearization is converted into a configuration dependent "gain" on input torques which accounts for the inertia change as the elbow angle changes. The outer loop PD controllers was designed using root locus. The outer loop MMLQR was designed by linearizing the arm with the mass matrix decoupler around two angles and then finding the LQR matrix. The two angles are 90 and 10 degrees as before in the regular MMLQR.

After the second downselect, the MMLQG, FBLQG, and GSLQR were chosen to be tested further, as well as the baseline PD controller. They were tested over a wide range of elbow joint configurations for the $FLEX_1$ simulation. When the $FLEX_2$ simulation was developed, the controllers were redesigned for the new payload and friction effects. The PD controller was also dropped in favor of the SRMS rate controller, an accurate model of the current controller on the SRMS. The controllers selected for comparison with the current SRMS rate controller were the robust LQG, the gain-scheduled LQG (GSLQG), and rigid arm feedback linearization with a robust LQG outer loop (FBLQG). These controllers were implemented in space-realizable form as outer loop controllers around the SRMS rate controller.

The selected four controllers were compared on the midsize payload FLEX model for a representative space construction task. The task consisted of removing a 7500-lb component from the Shuttle bay and positioning it for assembly. The weight represented an average of the payloads to be moved during ISSA construction, and is dynamically "heavy". That is, the separation between the lowest mode and higher modes is large. In order to avoid having a controller achieve good performance at one arm configuration at the expense of stability at another, each of the controllers is designed to achieve the best average performance across the entire SRMS workspace. This also ensures that each technique can indeed stabilize the arm in the presence of geometric nonlinearities.

Two facets of the move were considered: a large slew from the payload bay to the assembly area, and a small maneuver to position the payload for mating. The large slew was a straight vertical motion of the payload from 5 feet inside the Shuttle bay to 19 feet above the Shuttle. The primary metric was the time to settle within a circle of 2 inch radius, centered at the target position. This is based on the positioning accuracy requirement of the SRMS.

The mating maneuver was investigated by modeling the human operator's commands as small position commands (the elbow and shoulder pitch joints were commanded to move half a degree). The move was carried out around a series of nominal elbow angles (from 0 degrees to -160 degrees, the range of the SRMS elbow pitch joint) to assess the sensitivity of each controller to geometric nonlinearity. The results of feedback control will be presented in the next chapter.

3.6 Feedforward Methods

Feedforward techniques were compared on the basis of sensitivity to modeling errors, compatibility with teleoperation, complexity of the required calculations, amount of vibration reduction, length of command delay, and maturity of the technology. The following four techniques were considered: smooth trajectory generation, notch and lowpass command filtering, plant inversion, and input shaping. Feedforward methods which were considered are as follows: smooth trajectories, notch and lowpass filters, plant inversion, and input shaping. 'Smooth' trajectories, or minimum jerk command trajectories, limit energy put into a system to a band below the system's first natural frequency. Notch and lowpass filters are used regularly for signal processing and to get rid of vibrations at certain frequencies or beyond certain corner frequencies. Plant inversion works by modeling the system, inverting the model, and feeding the desired system response to the inverted plant. Then the resulting output becomes the command given to the actual plant. Input shaping has previously been described in Chapter 2.

Plant inversion and predetermination of smooth trajectories were dropped from consideration because they proved incompatible with teleoperation; they require that the input command be known before the move begins. Passing the commands through digital filters introduced long delays and limited system rise times, and was not consistently effective in reducing vibration. Input shaping has none of these drawbacks and has been shown to be effective for teleoperated systems, and so was considered in some detail.

The controllers are coded in FORTRAN and compiled into executables, and then there is a MATLAB shell program that generates the command and runs the executable. Input shaping was implemented by adding the appropriate code to the MATLAB shell. Since input shaping is a feedforward method, the only modifications necessary were to the input generation sections. The trajectories were known before the moves started, so input shaping was implemented after the commands were calculated. System frequencies and damping ratios, the information necessary to generate the shaped trajectories, were extracted from linearized models of the systems. Test moves were chosen to be the same as those used in the feedback tests. ZV and ZVD input shapers were chosen to represent the input shaping field. They were chosen because they were quick and easy to implement and do not excite higher frequencies or press saturation limits like the negative shapers do.

3.7 Conclusions

Methods for feedback and feedforward control of flexible structures and for robot teleoperation have been studied in depth by many researchers. The FLEX program builds upon this previous work but will have much more impact on space construction and maintenance. The technologies of nonlinear feedback control, robust control synthesis, and feedforward input shaping have been extensively explored. However, these explorations have been conducted somewhat in isolation. The FLEX team has combined feedforward techniques with robustified, nonlinear control in a manner which exploits the benefits of both.

During Phase A we evaluated an array of candidate feedforward and feedback control methods, and selected a few most promising ones for further consideration. Useful information was obtained by rating the performance of these control methods in completing a standard set of benchmark tests on the three-dimensional simulation model of the SRMS that we developed during Phase A. We have completed these sets of comparisons and consequently we understand a good deal about the capabilities of each control method.

FLEX Results

Chapter 4

This chapter will present the important results from Phase A of the FLEX program. This chapter explores the interactions between feedback control and feedforward methods. When you combine the two, does performance improve? By what percent does performance improve, a lot or a little? Can the feedback controller be redesigned with higher gains and then get better performance when input shaping is added? Do you get enough improvement by adding input shaping to justify the associated time delay?

There are many variables that could be considered. There are different types of feedback controllers, different feedforward methods, different duration moves, planar or 3-D moves, different payloads, different nonlinearities that can be added, and many more factors. Performance can also be evaluated based upon point-to-point criteria or trajectory following criteria. For input shaping, there are many variables such as number of modes, insensitivity of the shaper, and which frequency is shaped for. For each set of tests discussed, different variables were selected to be the axes of the test matrix.

4.1 Unloaded SRMS Results

4.1.1 Procedure

The planar model of the SRMS with a nominal payload and no coulomb friction was tested first. This model locks the shoulder yaw joint and just actuates the shoulder pitch and elbow pitch joints. For sake of brevity, the shoulder pitch will be called the shoulder joint and the elbow pitch joint will be called the elbow joint. This planar simulation has 18 states, 2 inputs, and 15 outputs and will be referred to as $FLEX_1$ in the rest of this chapter.

Four controllers have been tested extensively on $FLEX_1$: the PD, multiple-model LQG (MMLQG), gain-scheduled LQR (GSLQR) and feedback linearized LQG controller (FBLQG). Additionally, a sensitivity-weighted LQR (SWLQR), multiple-model LQR (MMLQR), a feedback linearized PD controller (FBL), and a feedback linearized MMLQR controller (FBLMM) were examined before being eliminated during the feedback controller downselect.

The precise positioning task discussed in Chapter 3 was investigated by modeling the human operator's commands as small step commands. The shoulder and elbow joint step commands started at 0.25 degrees below the desired angle and ended 0.25 degrees above the desired angle. Many small slew runs were done for each controller about three different elbow angles, 10, 50, and 90 degrees. After the feedback controller downselect, each controller was tested over a series of nominal elbow angles (from 0 degrees to -160 degrees, the range of the SRMS elbow pitch joint) to assess the sensitivity of each controller to geometric nonlinearity. θ_{elb} is the elbow angle.

The performance metric used is settling time. Settling time, t_s, is defined here as the time to settle to within a certain radius about the final desired position; it is a two-dimensional performance metric. For these tests the radius is equal to 2% of the total distance traveled or a maximum radius of one inch. For the small slews, the distance traveled by the joints is 0.5 degrees, which means that the tip travels 0.51 feet. The radius is 2% of that which equals 0.12 inches. This is a very small settling radius, but the small slews will be used by astronauts for precise positioning tasks where final location is very important. In the result tables other variables are the length of the input shaper delay, t_{is}, and the shaper frequency, f_1.

A variety of input shaping methods were examined, including single and multiple mode input shapers. ZV and ZVD shapers are the primary input shapers tested here. The system frequencies are calculated from the eigenvalues of the linearized system matrix. Each controller has different primary frequencies, so different input shapers are calculated for each controller at each configuration tested.

4.1.2 Results

The initial small slew results are shown in Table 4.1, Table 4.2, and Table 4.3. The PD controller was run for a single mode ZV and ZVD and several two mode shapers. After looking at these results it can be seen that the shortest duration input shaper is giving the largest improvement in performance. The ZVD and two mode shapers delay the response enough that the reduction in vibration amplitude does not fully compensate for the command delay. For this reason the other controllers were only tested with a single mode ZV shaper, though some controllers were tested at various shaper frequencies.

Different frequencies were tried because when the primary frequency has a high damping ratio, around 0.5, it is unclear whether shaping for that frequency will help or hurt the response. This was shown in Chapter 2 for a simple linear system. As the system damping approaches the critical level, input shaping should help performance less and less. Originally the runs were done for the second mode which has a higher frequency and lower damping ratio. However, there was very little or no improvement in performance. So the input shaper was recalculated for the first mode which had a lower frequency and a higher damping ratio. The systems settled faster when shaped for the lowest frequency, regardless of whether or not the damping ratio was approaching the critical damping.

controller θ_{elb}=10 delta=0.5	t_s unshaped (sec)	t_s shaped (sec)	f_1 (Hz)	Shaper	t_{is} (sec)
PD	5.27	1.29	0.42	ZV	1.24
PD	5.27	2.37	0.42	ZVD	2.48
PD	5.27	2.14	0.42/5.65	ZV/ZV	1.28
PD	5.27	2.29	0.42/5.65	ZVD/ZV	2.29
PD	5.27	2.32	0.42/5.65	ZVD/ZVD	2.56
SWLQR	2.87	1.24	0.43	ZV	1.37
SWLQR	2.87	2.88	4.14	ZV	0.14
MMLQR	1.95	1.07	0.49	ZV	1.23
MMLQR	1.95	1.97	3.86	ZV	0.14
FBL	2.24	1.43	0.65	ZV	0.82
GSLQR	1.70	1.69	3.57	ZV	0.14
FBLMM	1.62	1.61	6.89	ZV	0.07
SWLQG	4.02	2.00	0.32	ZV	1.75
SWLQG	4.02	2.92	0.32/1.81	ZV/ZV	1.86

Table 4.1: $FLEX_1$ results for small slew around θ_{elb}=10 degrees

controller θ_{elb}=50 delta=0.5	t_s unshaped (sec)	t_s shaped (sec)	f_1 (Hz)	Shaper	t_{is} (sec)
PD	4.01	1.21	0.46	ZV	1.15
SWLQR	2.04	2.06	4.11	ZV	0.13
SWLQR	2.04	1.20	0.47	ZV	1.20
MMLQR	1.80	1.84	3.79	ZV	0.14
MMLQR	1.80	1.05	0.54	ZV	1.16
FBL	1.92	1.44	0.70	ZV	0.76
GSLQR	1.78	1.82	3.62	ZV	0.14
FBLMM	1.57	1.58	5.81	ZV	0.09
SWLQG	3.54	1.84	0.36	ZV	1.58
SWLQG	3.54	2.60	0.36/1.68	ZV/ZV	1.69

Table 4.2: $FLEX_1$ results for small slew around θ_{elb}=50 degrees

The PD controller benefits the most from input shaping. Its settling time is reduced at least by half and at most by 75% when a ZV shaper is added. These shapers take out most of the vibrations and the time delay is very short. The results are not as clear for the more complicated controllers. The MMLQR, SWLQR, and SWLQG benefit when a ZV input shaper is added. The FBL, GSLQR, and FBLMM do the same or slightly worse when input shaping is added. They are very nonlinear controllers, so this is not unexpected. The feedback controllers are designed to add a lot of damping to the modes, so the lowest frequencies are critically damped. When the FBL and GSLQR inputs are shaped for the higher frequencies, their performance does not improve. The fastest settling time is achieved with the MMLQR and a ZV shaper. It settles in about 1.05 seconds, compared with a minimum unshaped settling time of 1.5 for the FBLMM controller.

controller θ_{elb}=90 delta=0.5	t_s unshaped (sec)	t_s shaped (sec)	f_I (Hz)	Shaper	t_{is} (sec)
PD	2.50	1.08	0.56	ZV	0.96
PD	2.50	1.82	0.56	ZVD	1.91
PD	2.50	1.70	0.56/20.42	ZV/ZV	0.91
PD	2.50	1.68	0.56/20.42	ZVD/ZV	1.8
PD	2.50	1.69	0.56/20.42	ZVD/ZVD	1.82
SWLQR	1.76	1.66	4.00	ZV	0.13
MMLQR	2.06	1.50	0.58	ZV	0.99
FBL	1.86	2.02	0.82	ZV	0.66
GSLQR	1.66	1.70	3.69	ZV	0.14
FBLMM	1.41	1.52	4.79	ZV	0.11
SWLQG	2.23	1.88	0.72	ZV	0.77
SWLQG	2.23	2.20	0.72	ZVD	1.54
SWLQG	2.23	1.84	0.72/1.92	ZV/ZV	0.95

Table 4.3: $FLEX_1$ results for small slew around θ_{elb}=90 degrees

The results show that the input shapers works best for the systems with the lowest frequencies and simplest feedback controllers. The MMLQR and SWLQR controllers do not shift the closed loop frequencies very far (though damping is added), so the input shaper can get rid of them easily. The GSLQR and FBLMM controllers add enough damping to the modes with frequency less than 1 Hz that the damping is above 0.7; therefore it does not make sense to shape for those frequencies. However, the next highest frequency (with a damping less than 0.7) is around 3.5 Hz for the GSLQR and 5.5 Hz for the FBLMM. There is not that much energy in these modes, so the input shapers end up adding a delay to the system. The SWLQG has a huge overshoot in the unshaped case. Here, input shaping really works well. But even with input shaping, we cannot get down to settling times achieved with MMLQR or PD plus ZV input shaping.

After these simulations were run and analyzed, a feedback controller downselect was performed. Three controllers were chosen to be tested further, the GSLQR, FBLQG, and MMLQG, as well as the baseline PD. The decision was made to change to LQG instead of LQR controllers in order to account for the limited number of sensors available for use on orbit. The final controllers were then run for a 0.5 degree step, for a range of elbow angles between 0 and 160 degrees. A single mode ZV and ZVD shaper were added to the inputs.

Figure 4.1 and Figure 4.2 show the individual controller performances for the unshaped and two input shaper cases. Unfortunately, the MMLQG proved only to be stable for a very small angle move and so was moved for delta angle=0.05 degrees. The PD, FBLQG, and GSLQR were moved for delta angle=0.5 degrees.

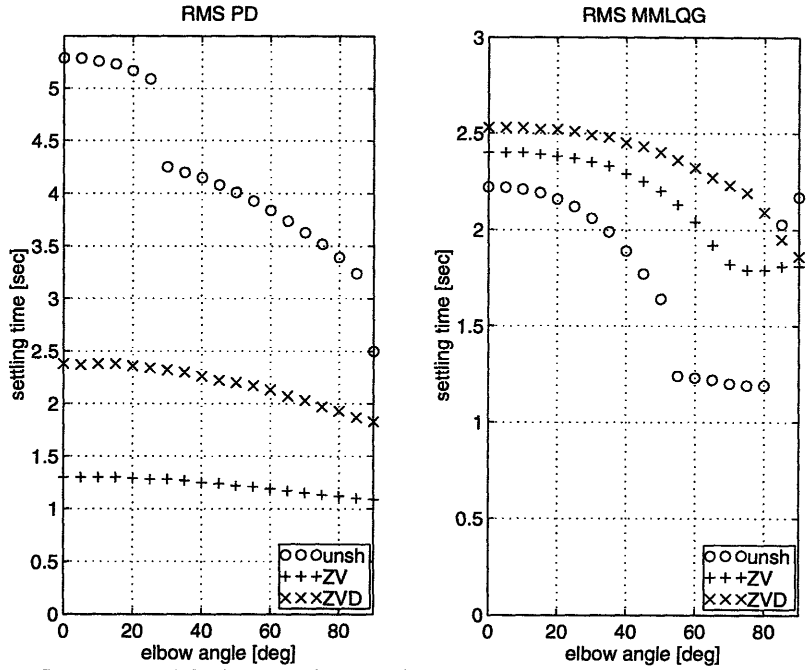


Figure 4.1: $FLEX_1$ simulations for PD and MMLQG controllers

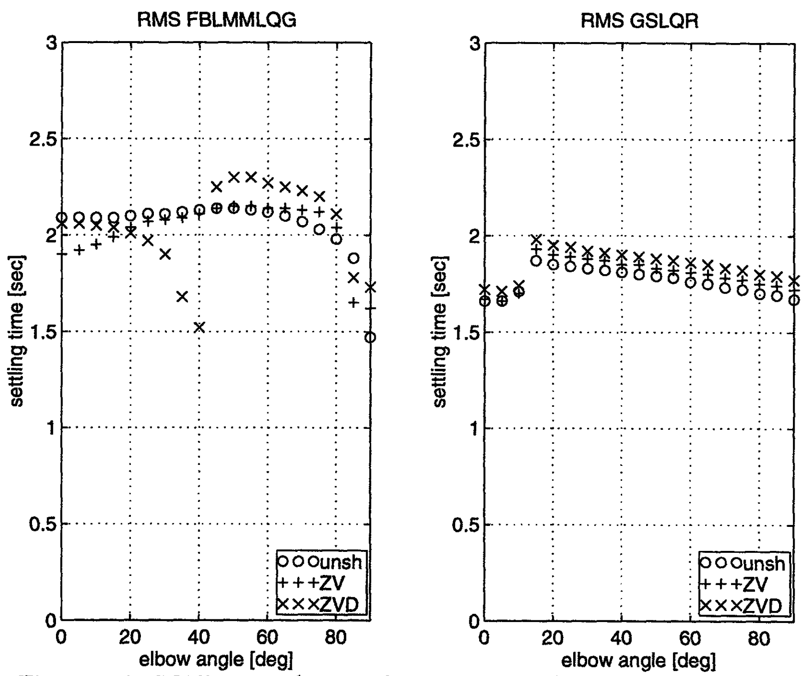


Figure 4.2: $FLEX_1$ simulations for FBLQG and GSLQR controllers

The PD controller does much better when input shaping is added. The ZV shaper gives the most performance increase because the time delay is half that of the ZVD shaper. But the ZVD shaper still does better than the unshaped step responses. The PD controller was designed for an elbow angle of 90 degrees, so its unshaped performance is the best at that point. However, by adding the ZV shaper, the performance is evened out over the whole range so comparable settling times are achieved everywhere. The MMLQG does not benefit from adding input shaping, unlike the original MMLQR. The GSLQG also is not helped by the addition of input shaping, though the input shapers do not add as much time delay as they did to the MMLQG.

The FBLQG is a puzzling case. Exact feedback linearization is a rigid robot technique where inverse dynamics is used to get rid of the rigid body modes. Then a MMLQG controller is added as the outer loop. This make the FBLQG system a very complex and nonlinear one. Input shaping was derived as a linear system technique, though it has been effectively applied to nonlinear systems. Here input shaping is not having much effect in the ZV shaper case and is both improving and degrading performance in the ZVD case. There is definitely some interaction going on between the feedback linearization method and the input shaping, but it is not clear exactly what is happening.

Figure 4.3 shows the unshaped results for the $FLEX_1$ simulation over the range of 0 to 90 degrees. The PD, FBLQG, and GSLQR results are moving a distance 10 times that of the MMLQG. Thus the MMLQG results are not comparable and it is doing decidedly worse than the others, taking just as long to travel one-tenth of the distance that the others travel.

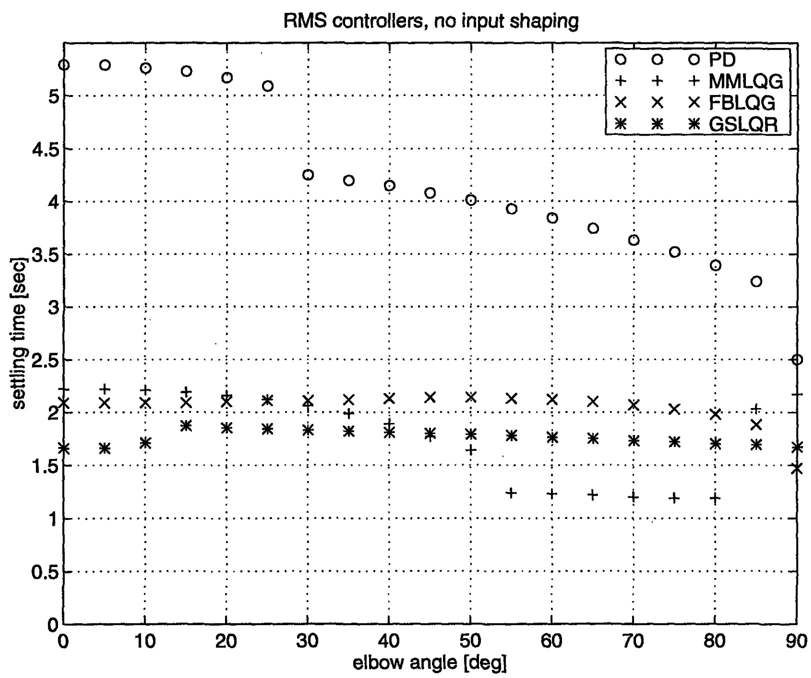


Figure 4.3: $FLEX_1$ unshaped results

Figure 4.4 shows the results for the ZV shaper. The PD plus ZV input shaper definitely does the best here. If results are compared with Figure 4.3, it can be seen that the PD plus ZV is settling faster than the best unshaped controller, the GSLQR.

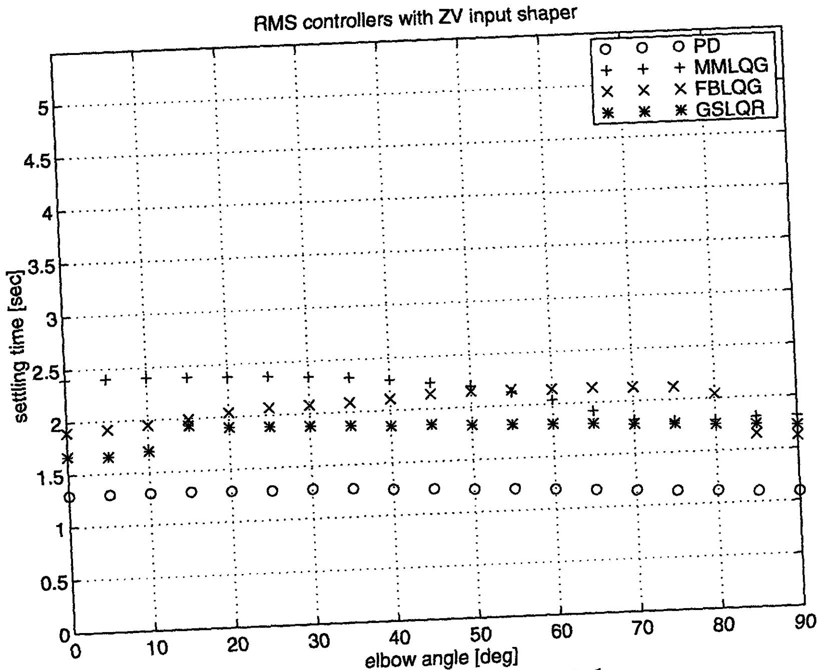


Figure 4.4: $FLEX_1$ results for a ZV shaper

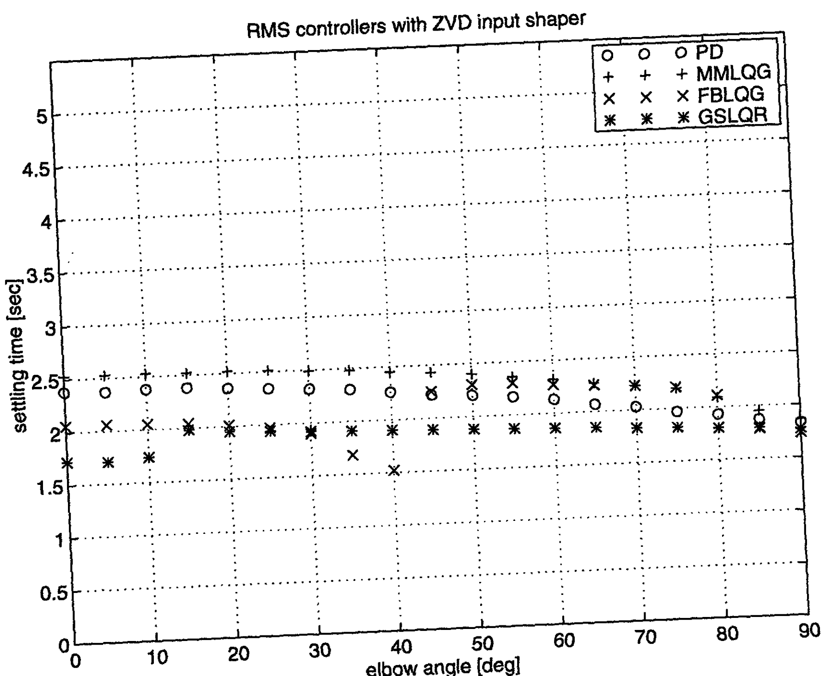


Figure 4.5: $FLEX_1$ results for a ZVD shaper

Figure 4.5 shows all of the controllers settling times when ZVD shaping is added. The interesting thing here is that the results are all very close to each other. Performance is evened out by adding the ZVD shaper, perhaps because the insensitivity of the shaper and the time delay allows the fundamental frequency enough time to die out for each controller. The results are not shown for the 90 to 180 degree range; however, they look very similar to the results shown here, and so are not enlightening.

4.1.3 *Conclusions*

Input shaping does increase the performance of the feedback controller if the controller is not too nonlinear. When input shaping was added to the PD, MMLQR, and SWLQR, performance noticeably increased. However, when input shaping was added to the more nonlinear controllers such as FBLQG and MMLQG, the performance stayed the same or was degraded.

The PD controller plus ZV input shaper did better than all of the unshaped system responses. It has the additional benefit of being easy to implement and does not require much computational resources or any additional sensors. These results are for the SRMS model with no payload and no friction, both of which make the system and strategies more complicated.

4.2 SRMS Proportional Controller

4.2.1 *Purpose*

One of the major questions raised in the previous study was what happens when you relax the settling performance constraint and allow the feedback controllers to move faster. How much more quickly do you reach the desired position? We are trying to assess the value to be gained by adding input shaping. Can the system reach its goal faster if shaping is used? We have learned how quickly the system can accomplish its move with a well damped PD controller. Now we are trying to learn whether we can do better with a proportional controller (with a very small derivative gain).

The decision was made to do a series of tests on the PD controller to see what the interactions were between the feedback controller and input shaping. Usually the feedback controller is designed first and input shaping is added later. If they are designed concurrently, theoretically you could get a large performance increase.

4.2.2 *Test Protocol*

These questions were investigated by modifying the $FLEX_1$ plus PD controller model. Motor saturation was taken out of the model in order to simplify the problem. The $FLEX_1$ code was also modified so that it reads in the proportional and derivative gains for the elbow, *Kpe* and *Kve*, from a MATLAB file. The shoulder gains were kept equal to the original PD controller gains. The FLEX simulations for this section will be called $FLEX_{1m}$.

One of the first decisions to be made was what performance metrics to use. The rise time was chosen to be one metric. The radius rise time was defined to be the first time the tip position enters the settling radius. However, with proportional and derivative control, sometimes the tip misses the settling/rise radius on the way up, and on the way down. It is swinging past the desired position without entering the settling circle. In fact, it often misses it for many swings back and forth. In order to circumvent this problem, elbow rise time was chosen to be the time it takes for the elbow joint to complete two-thirds of the commanded move. Both radius and elbow rise time were calculated for the runs, as well as the settling time.

Initially I was going to study just a proportional controller. However, with no derivative gain , the controller went unstable for a low gain. By adding a very small derivative gain of *Kve*=0.0010, the maximum desired proportional gain of *Kpe*=0.30 was stabilized. *Kve* is only one order of magnitude below the lowest proportional gain of the study, *Kpe*=0.01. The original gains that were designed for the elbow are *Kpe*=0.05 and *Kve*=0.0163. By adding the small amount of damping, the system poles move off the imaginary axis. In practice, every mechanical system has some damping, so this is a realistic addition. When the poles are on the imaginary axis in the simulation, we see unstable higher modes which would not become unstable in practice.

Runs were done for a shoulder angle of 0 degrees and a nominal elbow angle of 45 degrees. The unshaped input to the system was an elbow step command from 40 degrees to 50 degrees. The tip of the arm moves four feet during the move. The settling radius is one inch around the desired final tip position. Runs were done for a range of proportional gains between 0.01 and 0.3. All runs were done for an elbow proportional gain of *Kve*=0.0010. For the shaped runs, different input shapers were tried until the vibration was reduced enough so that the shaped response settled at the same time it reached the rise radius. This was done to see how much input shaping was needed for each proportional gain and to see if it varied along the test matrix.

4.2.3 *Results*

The rise time results are shown in Figure 4.6. The radius rise time to a 1 inch radius about the final desired position is shown in the first subplot. This metric does not give consistent results; sometimes the tip misses the rise radius until it has vibrated for a while. The second subplot shows the elbow rise time; these results show a more consistent pattern. The shaped elbow rise time is always greater than the unshaped rise time. This shows the effect of the time delay introduced by the input shaper.

The reason why the rise times are higher when the gains are less than 0.04 is that different shapers are needed for the lowest gains. The first two shaped points, *Kpe*=0.01 and 0.02, needed single mode ZVD shapers to get the settling time to equal the radius rise time. The second two points, *Kpe*=0.03 and 0.035, needed single mode ZVDD shapers to get the settling time to equal the rise time. At *Kpe*=0.04 the system switches to needing two mode, ZV/ZV shapers for the rest of the cases and the curve levels off. One reason why this is happening is that the dominant mode is not changing frequency as much after Kpe=0.04, which is why the results look the same.

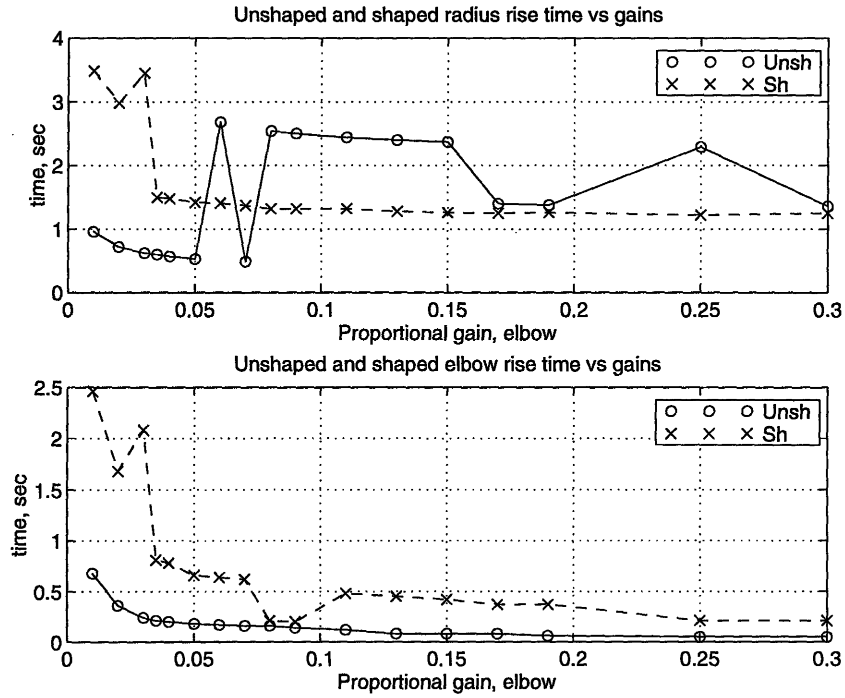


Figure 4.6: Unshaped and shaped rise times for $FLEX_{1m}$

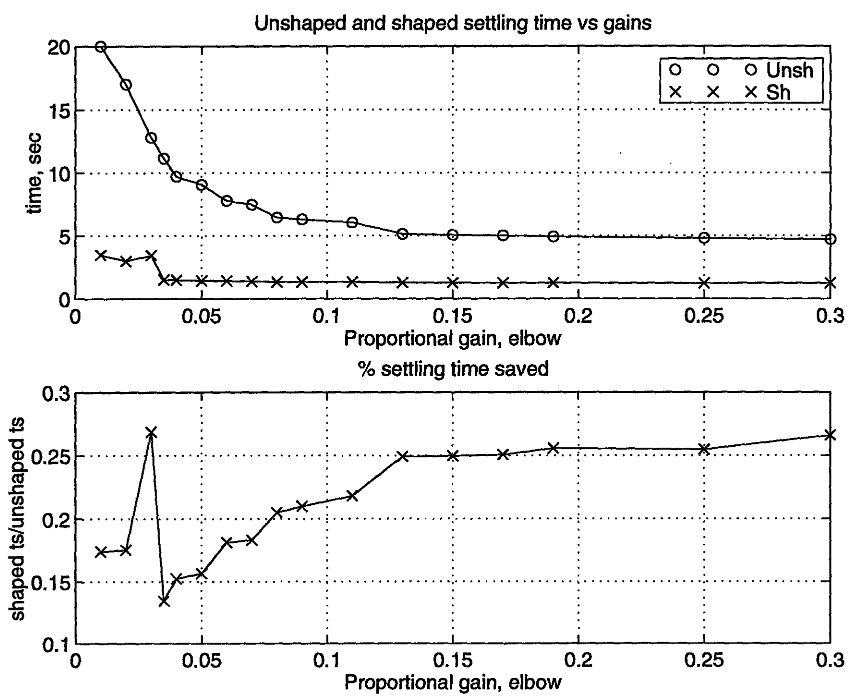


Figure 4.7: Unshaped and shaped settling times for $FLEX_{1m}$

Figure 4.7 shows the settling times for the shaped and unshaped responses. The settling times level off at a proportional gain of about 0.15. So there is a minimum settling and rise time below which the system cannot go, despite the lack of friction and saturation. The second subplot in Figure 4.7 shows the percentage of time saved, the shaped settling time divided by the unshaped settling time. It is decreasing with proportional gain, as the unshaped settling time gets smaller and smaller. However, the shaped case always saves you time.

One reason why the response levels off after a certain gain is shown in Figure 4.8. The first mode reaches a certain frequency and levels off, while the second frequency continues to increase with increasing proportional gain. The dominant pole is approaching a zero and thus cannot change any more.

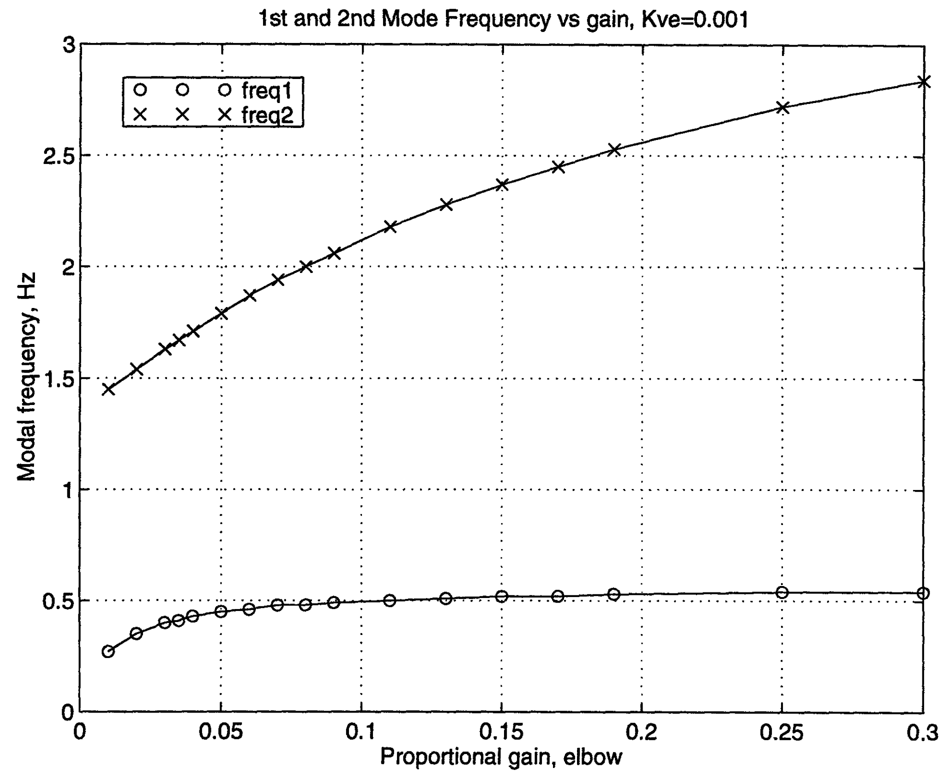


Figure 4.8: First and second mode frequencies for $FLEX_{1m}$

4.2.4 Conclusions

It is not as easy as one would like to increase a controller's performance. In this case we were trying to increase the system rise time by increasing the proportional gain of a PD controller. This worked very well for low gains, where a significant difference can be seen. However, the system constraints meant that increasing the system gains past a certain point had no effect. The fundamental mode approached the corresponding zero and the frequency of the system could not rise any higher. The original proportional gain of the PD system was 0.05, so the PD controller was designed very well. By decreasing the derivative gain, the system did move faster and input shaping was able to make the system settle faster than before. This study leads me to believe that it is hard to design feedback controllers and input shapers in concert unless you have a very well known system that behaves well. And even if you do know the system very well, it will still be hard to design the feedback and feedforward systems.

4.3 Midsize Payload Results

4.3.1 *Test Matrix*

Once the FLEX model was redesigned to accept payloads and the friction level was raised, the controllers had to be redesigned. A sample payload of 7500 lb. was used to represent an average of the payloads to be moved during space station construction. This payload meant that the system frequencies changed from around 0.5 Hz to 0.09 Hz. This difference was too large for the original controllers to handle, so they were redesigned to be stable and effective in the new workspace. The controllers are the SRMS rate, which uses same gains as the controller on the SRMS, the LQG, the FBLQG, and the GSLQG. This model of the FLEX simulation will be called $FLEX_2$.

The new controllers were tested on a representative space construction task. The task consisted of removing a 7500-lb component from the Shuttle bay and positioning it for assembly. This task was broken into two parts: a large slew from the payload bay to the assembly area, and a small maneuver to position the payload precisely and accurately. The large slew was a straight vertical motion of the payload from 5 feet inside the Shuttle bay to 19 feet above the Shuttle. The motion followed a trapezoidal velocity profile for the Z tip. The move was translated into the corresponding joint position and velocity commands, which were fed to the $FLEX_2$ model. The primary metric was the time to settle within a circle of two inch radius, centered at the target position. This requirement is based on the positioning accuracy requirement of the SRMS.

We assumed that settling time is a primary limitation to productivity, and that rise time plays a role in perceived arm responsiveness. Therefore controllers were evaluated by comparing average rise times and average settling times at all of the nominal angles.

4.3.2 *Results*

Large slew settling times for each of the controllers are shown in Table 4.4 (as t_s). Also recorded were the average and maximum end-effector position errors (e_{avg} and e_{max} respectively). The large slew settling time for the SRMS rate controller is not reported. The rate controller does not incorporate position feedback; it relies on astronaut input to reach a desired position. As a result, settling time in response to a position command is not a relevant metric. (The arm does reach the desired position, but not in a time comparable to that possible under human control). The implementation costs of each controller were assessed using the following metrics. The total energy used in the large slew was calculated (E). The number of lines of code in each controller was determined as an indicator of the cost of validating the controller software. Finally, as an indicator of required real-time processing, the total control calculation time, t_c, for the large slew (80,000 cycles at 500 Hz) is shown. The rate controller does not have a number of lines or tc given because it is the default. All of the other controllers are outer loops around the rate controller.

The small slew results are also reported in Table 4.4, where t_r is rise time and t_s is settling time to a radius equal to 2% of the total distance traveled. The settling and rise times shown in the tables are the average times averaged over the entire range of nominal elbow angles from 0 to -160 degrees. For the small slew, SRMS controller results can be reported, since the rate commands can be scaled to produce the correct change in position.

Controller	Large Slew			Small Slew		Cost		
	t_s (sec)	e_{avg} (in)	e_{max} (in)	t_r (sec)	t_s (sec)	E (W)	# of lines	t_c (sec)
SRMS rate	*	7.2	13.5	6.15	37.9	1961	-	-
LQG	85.4	1.4	3.2	5.96	11.6	916	181	8.6
FBLQG	69.8	.9	4.2	4.33	25.7	923	212	11.5
GSLQG	67.8	.3	.8	2.37	4.2	612	512	34.0

Table 4.4: Evaluation of advanced feedback control on $FLEX_2$

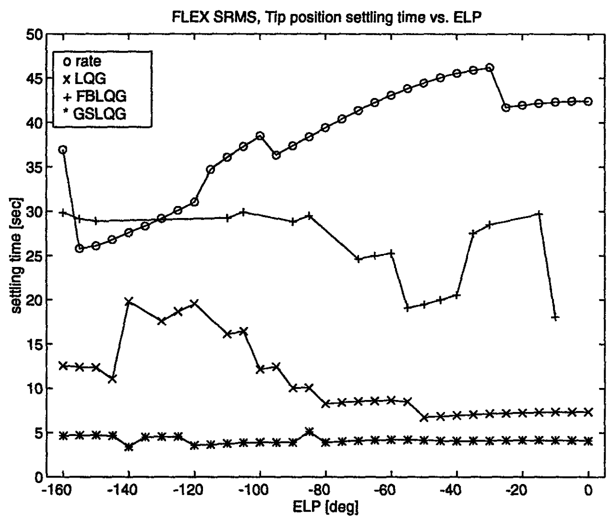


Figure 4.9: $FLEX_2$ settling times vs. elbow angle, advanced feedback control

Figure 4.9 shows the settling times, as a function of elbow angle, for the advanced feedback controllers, compared to the nominal rate controller. The best controllers achieve small settling times at all elbow angles. LQG achieves about three times better settling time than rate control, but loses authority around an elbow angle of -120 degrees. GSLQG settles almost ten times faster than rate control over the entire workspace. Comparison with Table 4.4 shows that, as expected, good performance is achieved at the expense of increased complexity, measured by both computation time and code size. The feedback-linearizing controller, FBLQG, does not include local loops at the joints to minimize the effects of Coulomb friction. Our experience suggests that adding such loops would significantly enhance the performance of this controller for small motions. The metrics for the long slew show that performance increases monotonically for increased control complexity, while energy used in the move decreases.

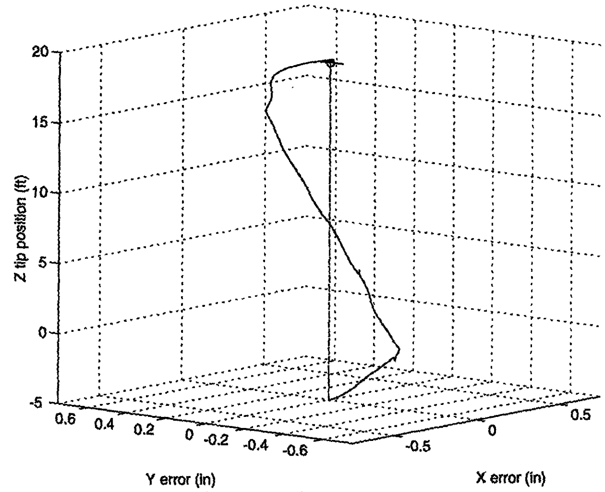


Figure 4.10: $FLEX_2$ payload path errors during the vertical move

Figure 4.10 demonstrates the performance of the arm under GSLQG control. The deviation (in inches) of the payload center of mass from the desired path during the large slew is plotted. The 7500 lb. mass travels 24 feet upwards in 60 seconds. The payload remains within an inch of the desired position throughout the move.

Table 4.5 shows the values for the same controllers and performance metrics defined above when input shaping was added to the input. The costs metrics are slightly different than before. The # of lines and t_c metrics refer only to the additional cost incurred by adding input shaping. The energy metric (E) is computed the same way as in the previous table. Once again the SRMS rate controller does not settle because of the lack of position feedback. However, input shaping did not noticeably improve the rate controller's performance.

Controller	Large Slew ZV			Small Slew ZV		Small Slew ZVD		Cost (due to ZV input shaping)		
	t_s (sec)	e_{avg} (in)	e_{max} (in)	t_r (sec)	t_s (sec)	t_r (sec)	t_s (sec)	E (W)	# of lines	t_c (sec)
SRMS Rate	*	10.2	26.9	6.13	9.08	8.83	26.5	1200	4	1.8
LQG	68.8	2.53	6.93	7.43	7.82	9.45	9.60	507	4	1.8
FBLQG	67.8	1.36	4.23	8.78	34.0	11.9	32.7	2335	4	1.8
GSLQG	67.4	1.22	3.33	3.53	3.43	4.52	4.42	1095	4	1.8

Table 4.5: Evaluation of advanced feedback with input shaping on $FLEX_2$

For the large slews, input shaping helped the settling time, but increased the average and maximum errors associated with trajectory following. This is due to the time delay introduced by the shaper. The shaped response lags the trajectory, and so incurs larger errors even though it follows the path more closely than the unshaped response. The unshaped commands end at 70 seconds and the FBLQG and GSLQG responses settle before the command ends for both the unshaped and the ZV shaper cases. Thus, there was not much that the input shaper could do to improve these controllers' performances. However, the input shaper did not add a noticeable delay to the large slews. This is because the duration of the input shapers was short compared to the move duration. The trapezoidal velocity command given to the controllers seemed to minimize vibration; the unshaped responses did not vibrate much after the move was done. Even the rate controller did not vibrate much; it just didn't settle quickly because of the lack of position feedback.

For the small slew moves, the rise times for each controller have all increased or remained the same, due to the time delay added by the shaping of the input. The ZV input shaper achieves the largest improvement with the rate controller. Average settling times drop from 38 seconds to 9 seconds. The ZV input shaper also improves performance when coupled with the LQG and GSLQG controllers. The nonlinear friction associated with FBLQG contributes to its poorer performance when used with input shaping. The total energy increases when input shaping is added to the FBLQG which indicates that the nonlinearities inherent in the FBLQG are hindering the input shaper's performance. Figure 4.11 shows the settling times, as a function of elbow angle, for the combined advanced feedback controllers and the ZV input shaper.

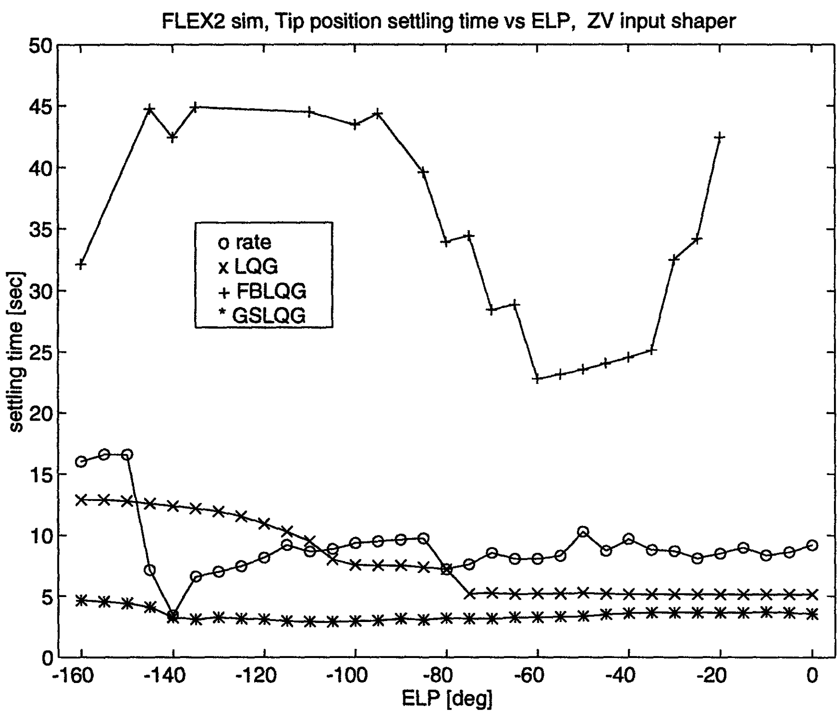


Figure 4.11: $FLEX_2$ settling times, advanced feedback with ZV input shaping

The ZV input shaper does improve the LQG, GSLQG, and SRMS rate performance for small slews. By contrast, the ZVD shaper does not improve the performance of the GSLQG; it just delays the settling time. The ZVD shaper does improve the performance of the LQG and SRMS rate controller, but not as much as the ZV shaper does. Interestingly, the ZVD input shaper does not degrade performance quite as much for the FBLQG when compared with the ZV. However, they still both do worse than the unshaped FBLQG which settles about 8 second faster than the ZVD shaped responses. Figure 4.12 shows the ZVD settling responses over the tested elbow joint range.

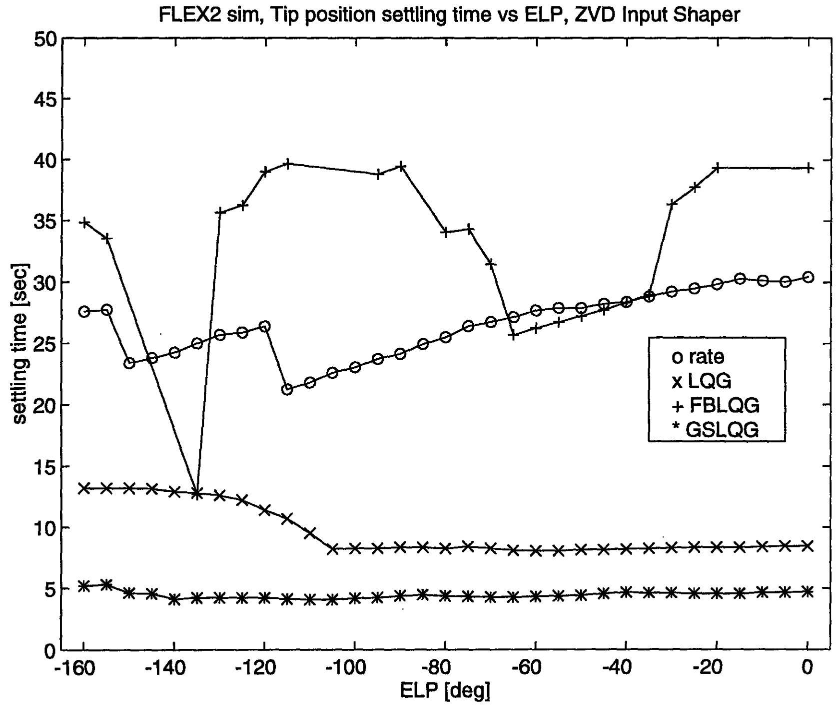


Figure 4.12: $FLEX_2$ settling times, advanced feedback with ZVD input shaping

The GSLQG curve looks very similar to the unshaped GSLQG curve shown in Figure 4.9. The LQG curve is very smooth as well. The SRMS rate controller plus ZVD curve is much higher than the rate and ZV curve. The additional five second delay associated with the derivative constraint is causing the response to rise more slowly and seems to be exciting more vibration. The tip responses of the SRMS rate controller are shown in Figure 4.13. There is a lot of vibration present in the unshaped response, though the damping is high enough to get rid of the ringing within five cycles. The ZV response settles very quickly and immediately, without exciting any vibration. The ZVD response is slower and still has a small amount of residual vibration after the command finishes.

Figure 4.14 shows the FBLQG response for an elbow angle of 45 degrees. The ZV and ZVD input shapers are not helping reduce vibration at all. They are merely delaying the rise time and settling time. The shaped responses do not overshoot the desired position by as much as the unshaped, but do have as much residual vibration as the unshaped responses. The lack of modeled coulomb friction in the FBLQG feedback loop could be one cause of this performance degradation. FBLQG combines feedback and feedforward techniques already by modifying the input by multiplying it by the inverse dynamics. Putting additional feedforward techniques on top of this is obviously not helping the system performance.

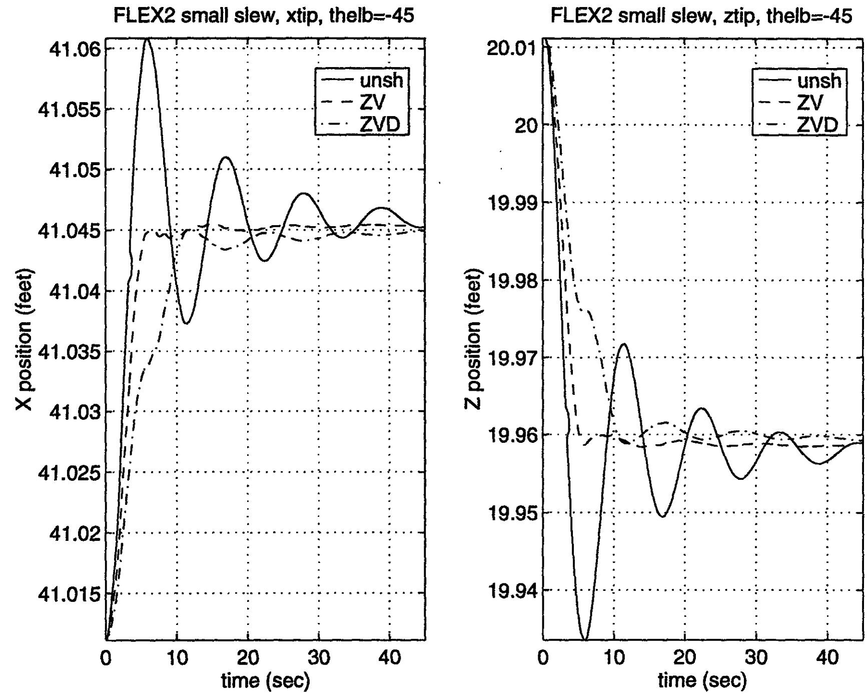


Figure 4.13: FLEX$_2$ SRMS rate controller tip responses for θ_{elb}=-45 degrees

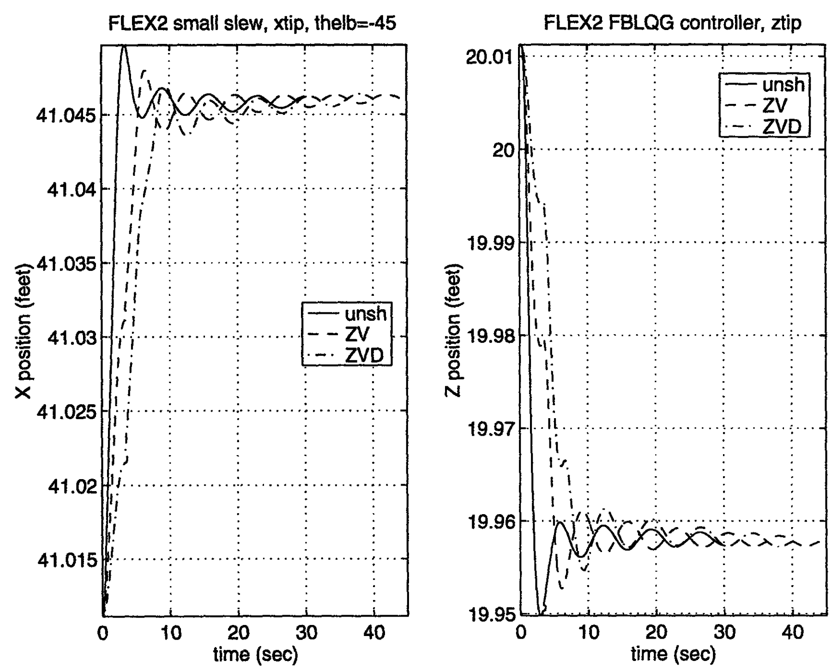


Figure 4.14: FLEX$_2$ FBLQG controller tip responses for θ_{elb}=-45 degrees

4.3.3 Conclusions

While many details of advanced control implementations on the SRMS and SSRMS remain to be addressed, the above study supports the following conclusions. First, the potential for significant performance improvement under advanced control exists. As much as a factor of six reduction in settling time, and a factor of 24 reduction in path-following error, are achieved in our simulation studies. Second, our cost metrics suggest that higher performance generally incurs higher costs. However, the feedforward control achieves a factor of four improvement over RMS rate control alone, with relatively minimal cost. This further suggests that cost and performance can be traded off by proper choice of algorithm, given a comparable set of performance and cost data for the algorithms.

4.4 Large Slewing Moves

A small study was done to investigate the $FLEX_2$ 2D trajectories. The move shown in the previous section was from z_i=5 to z_f=-19 feet. A variety of shorter and longer moves with z_i=5 and z_f=[-11 -15 -19 -23] were examined. The results are shown in Table 4.6. *tce* is the length of the command. The metrics have all been defined in previous sections. The GSLQG got rid of all of the vibration, no matter what length the move was. In fact it settled 3.4 seconds before the command reached its final point in all cases. The LQG controller also had good performance; though the settling time did increase by 0.2 seconds for each additional four feet moved. It settled from 3.4 to 2.6 seconds before the command reached its final point. The Rate controller had the worst performance; this is not surprising since settling time is a poor criterion here. However, it was not vibrating once the command ended, just settling very slowly due to the SRMS integrator.

Controller	Zfinal (feet)	tce (sec)	t_s (sec)	t_s-tce (sec)	e_{avg} (in)	e_{max} (in)	E (W)
Rate	-11	50	70.4	20.4	2.99	9.09	1398
Rate	-15	60	85.2	25.2	3.64	10.28	1493
Rate	-19	70	96.6	26.6	4.14	10.99	1612
Rate	-23	80	103.8	23.8	4.40	11.13	1725
LQG	-11	50	46.6	-3.4	0.94	3.52	761
LQG	-15	60	57.0	-3.0	0.88	3.52	698
LQG	-19	70	67.2	-2.8	0.92	3.52	714
LQG	-23	80	77.4	-2.6	1.00	3.52	831
GSLQG	-11	50	46.6	-3.4	0.18	0.73	877
GSLQG	-15	60	56.6	-3.4	0.19	0.73	1051
GSLQG	-19	70	66.6	-3.4	0.21	0.73	1180
GSLQG	-23	80	76.6	-3.4	0.24	0.73	1301

Table 4.6: $FLEX_2$ large slews

The fundamental frequency of the system was not changing since the shoulder joint was moving different distances, not the elbow joint. (The RMS is symmetric about the shoulder, so only the elbow angle determines the frequency.) The point of this test was to see if vibrations died out as a function of move distance. The answer appears to be that the vibrations were not apparent for any of these particular runs. Either they have all died out since the move lasted longer than 5 cycles of vibration, or the controllers get rid of them. So to get vibration, moves of much shorter duration or moves that change the system frequencies more must be investigated. Trapezoidal velocity trajectories are doing a very good job of reducing vibration for these long moves. The next thing to do is to change the velocity and accelerations of the velocity profile and see how much faster the system can move. This problem of optimizing performance for systems with changing geometries will be explored further in the next chapter.

4.5 Conclusions

The FLEX program successfully developed a number of SRMS models. Feedback controllers of varying complexity and effectiveness were developed for each model. The GSLQG controller performed very well on the midsize payload model with friction. It had the lowest settling times and the lowest trajectory errors. However, it also had the highest complexity and needed additional time in order to compute the controller at each time step. In contrast, the SRMS rate controller worked well when input shaping was added with little additional complexity or time penalty. However, the settling times were not as low as the GSLQG controller and the lack of position feedback impaired the trajectory settling performance. There is definitely a tradeoff between complexity and performance.

When an astronaut is added to the control problem, the results grow even more complicated. The interactions between an automatic feedback controller and an astronaut need to be investigated further, preferably on orbit. This will be investigated further in the later phases of the FLEX program.

Geometrically Varying Systems

Chapter 5

5.1 Introduction

Robotic manipulators are an example of a class of systems whose geometries change as they move. As the robotic arm travels through its workspace, the fundamental frequencies shift. This means that the feedback controllers must be stable and effective over a larger range of frequencies than the controllers for other types of systems. When input shaping is added to these systems, certain questions arise. Which frequency do you shape for? Potentially you could shape the input using the beginning frequency, the end frequency, the midpoint frequency, or using an adaptive input shaping method. There is also the general problem of how to deal with systems whose frequencies change with geometry and the more specific problem of how input shaping and feedback controllers interact while the geometry is changing. In this chapter, I will investigate the effects of input shaping with simple feedback controllers on geometrically varying systems.

The Space Shuttle Remote Manipulator System (SRMS) is a very good example of a changing geometry system. It is a fifty-foot long, six degree of freedom robot arm designed for use in space, since it cannot lift itself in a gravity environment. It deploys payloads from the orbiter's cargo bay and captures payloads for retrieval or repair. The SRMS can also be used as a platform for the astronauts during their extravehicular activities. It has seven joints but usually six degrees of freedom since the swingout joint is normally fixed during operation. Since it is very long and light, less than 1000 pounds, the SRMS is extremely flexible, especially when moving heavy payloads.

Singer examined geometrically varying systems in his doctoral thesis. Three types of systems were considered: systems for which the period of oscillation varies by a small amount, systems that experience large changes in frequency, but are velocity limited, and systems which experience large changes in frequency but are not velocity saturated during the move. For systems whose frequencies change less than 20% or 30%, he suggests that a more insensitive shaper is all that is needed. Systems which are velocity limited can have their accelerations and decelerations shaped to reduce residual vibrations. Systems with changing frequencies can be assumed to be "quasi-static" and use adaptive input shapers that change frequency at discrete points along the move.

5.2 Models

Tests were performed using several different models. A simple two-link model, based upon the SRMS, was created in order to get results quickly. Since the model has very few nonlinearities, the results should also be easier to interpret. The $FLEX_2$ simulation, which includes the variable payloads and friction effects, was tested further and has already been discussed in Chapters 3 and 4. The model that I explored the most was the DRS; it is known to be accurate and it captures many of the crucial nonlinearities such as friction, stiction, saturation, and backlash.

The decision was made to only move in single plane of the SRMS workspace, the *xz* plane. The planar assumption does simplify the problem. However, there are certainly enough nonlinearities present in the SRMS models to make the problem very interesting and hard to understand. The SRMS 2D model is a sufficient representation of a real system for certain cases.

5.2.1 *Two-link Model*

The two-link model was designed to represent a simplified SRMS. The system was modeled as a two-link flexible manipulator with a payload. The resulting model, referred to as the 2LM, consists of a fixed base joint, a rigid link, an elbow joint, another rigid link, and a wrist and payload. The joints are flexible and damped. The model is shown in Figure 5.1.

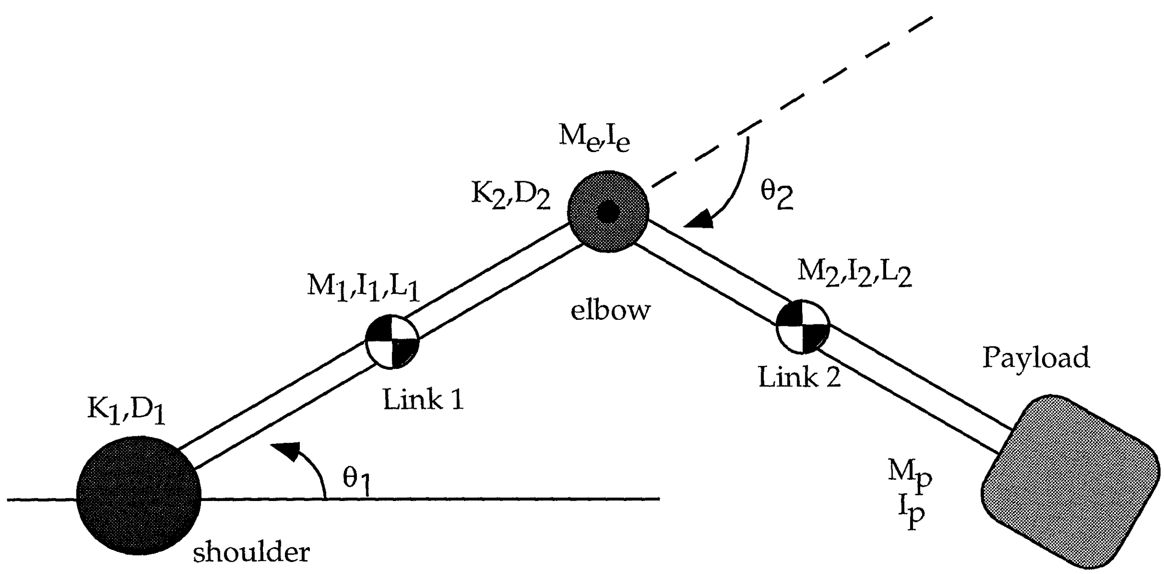


Figure 5.1: Two-link model and its parameters

The equations of motion for this system are given in the Appendix B. The masses, inertias, and link lengths were chosen to be equal to the actual SRMS values. The values are given in Table 5.1 and are from reference [19].

	Length (feet)	Mass (slugs)	Inertia (slug-ft^2)	Joint Stiffness (ft-lb./radian)
Link 1	20	8.0	266.0	
Link 2	23	3.5	154 2	
Shoulder Joint				2.11E6
Elbow Joint		5.2	0.4362	1.90E6
Payload		238.7	7063.5	

Table 5.1: SRMS parameters used in the two-link model

The stiffnesses and dampings of the joints were chosen to match the SRMS frequencies and to match the damping and settling times. Once the SRMS parameter values were added to the two-link model, its system response started to resemble the SRMS's response. The gearbox stiffness of the SRMS shoulder pitch is Kg_1=2.11E6 ft-lb./rad and the gearbox stiffness of the elbow pitch is Kg_2=1.9E6 ft-lb./rad, as given in Table 5.1. These stiffnesses were used as the baseline spring gains. The stiffnesses gave frequencies of the right order of magnitude, but were too high since some nonlinearities were missing from the two-link model. The SRMS rate controller has equal gains on each joint, so K_2 was chosen to equal K_1. When the spring gains were increased to K_1=2.6*Kg_1=K_2 and damping was added to model the rate controller, B_1=1.2E5 ft-slug-sec/rad and B_2=1.0E5 ft-slug-sec/rad, the two-link model became a very good representation of the SRMS. This factor of 2.6 that was necessary to get the frequencies to match can be considered a correction factor. There are many nonlinearities which were not included in this simple model. The correction factor is necessary to compensate for the missing elements. The correction factor was found by after trying many combinations of stiffnesses and damping ratios.

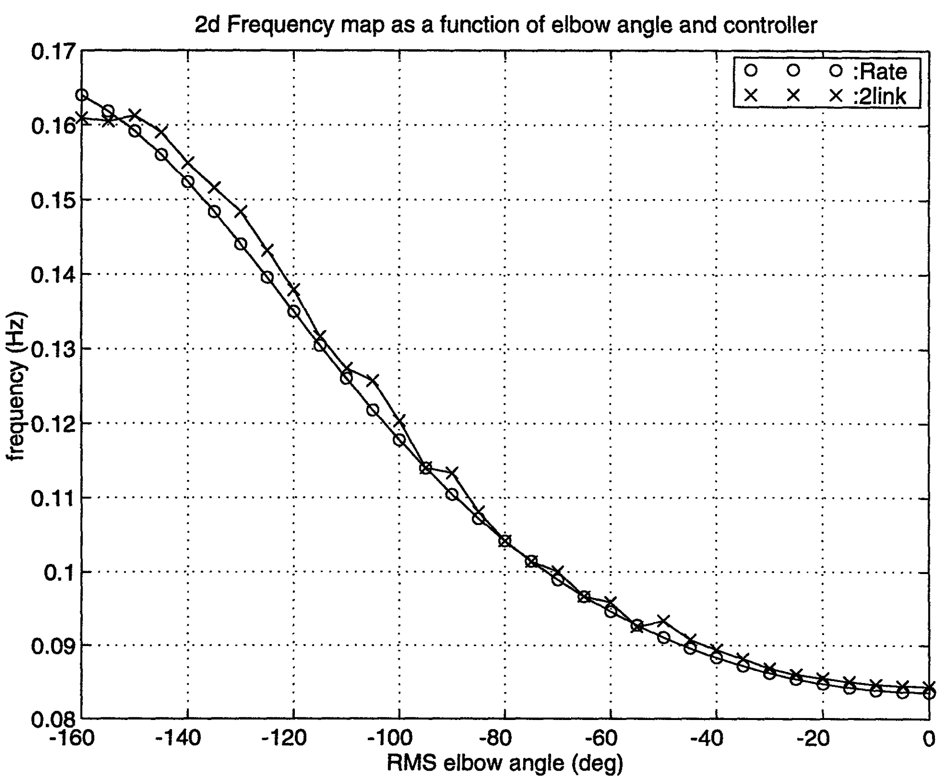


Figure 5.2: $FLEX_2$ model frequencies vs. two-link model frequencies

Figure 5.2 shows the $FLEX_2$ system frequencies versus the two-link model frequencies. This $FLEX_2$ system uses the SRMS rate controller as its feedback controller and both models use the 7500 lb. payload. There is good agreement (within 2.5% of each other) and both follow the same curve. The frequencies do not change as the shoulder angle is changed. Part of the reason why the joint stiffnesses and dampings were changed was to compensate for the lack of friction and saturation in the two-link model. Both of those nonlinearities could have been added, but they would have increased the run time, which was the reason for the simpler model. The model could also have been improved by incorporating torsional springs in the links, and placing a torsional spring under the base. However, adding torsional springs would only add two extra modes and would not necessarily clarify the problem. Currently the model has flexibility only in the joints and rigid links. The two-link model was able to match the first two frequencies of the SRMS by tuning the joint flexibilities and dampings, but does not model any of the flexibility in the links. Therefore, the two-link model results will never exactly match the more complicated models. It is a good first-order approximation, however.

5.2.2 *The Draper Remote Manipulator Simulation (DRS)*

The Draper Remote Manipulator Simulation (DRS) was developed by Draper Labs over a period of 10 years and designed to account for many nonlinear effects. It was extensively verified using actual flight data and was validated by NASA for use as a tool for analysis of shuttle payload deployment and retrieval operations. It has a flexible payload module to allow higher fidelity simulation of the internal motion of payloads with flexible components. [21]

Payload	Joint velocity limits (deg/sec)			
	Shoulder Yaw	Shoulder Pitch	Elbow Pitch	Wrist
unloaded	2.29	2.29	3.21	4.76
7500 lb.	0.51	0.51	0.71	1.06
32000 lb.	0.229	0.229	0.321	0.476

Table 5.2: Joint velocity limits for different payloads

The DRS simulation I used had been previously transferred to the UNIX environment by Singer. [25] The model includes saturation, motor resistance, motor coulomb friction, motor stiction friction, joint brakes, joint friction, gearboxes, and servo loops. Many different parameters can be changed, such as the friction, stiction, integral gains, sampling rates, payloads, inputs, outputs, etc. I used the default DRS parameters, as defined in Payload Deployment and Retrieval System Simulation Database, Version 1.0, but changed the payload parameters to model the various payloads correctly. [19] Three payload configurations were tested: the nominal payload (just the wrist and end-effector), a 7500 lb. payload used in the FLEX simulations, and a 32,000 lb. cylinder payload defined in the DRS reference manual. [16] Each of these payload has different characteristic joint rate limits, as stated in Table 5.2. These limits constrained the workspace by limiting the joint velocities. The motor torque limits restricted the allowable accelerations.

5.3 Procedure

5.3.1 Hypothesis:

There are different combinations of geometrically changing systems and moves:

I. Insensitive system: frequency changes less than 10%
II. Mid-sensitive system: frequency changes between 10% and 20%
III Sensitive system: frequency changes more than 20%
  A. move duration is longer than N cycles of vibration (of the fundamental frequency)
  B. move duration is less than N cycles of vibration

Each type of system needs a different solution. For example, for system I, a ZV or ZVD input shaper should remove the most vibration with the least time penalty. They are faster than the more insensitive shapers and should give the greatest improvement with the least cost unless there is a large error in system knowledge. System II needs a more robust input shaper, such as a ZVD, ZVDD, EI, or EI two-hump. The sensitivities and time delays associated with each of these input shapers are given in Table 2.1. N is defined to be the number of cycles at which the initial acceleration is no longer affecting the residual vibration. N is different for each system, and so is not chosen to be a specific number here. System damping plays a large part in defining N; the higher the damping, the lower N should be.

System III, the very sensitive system for which the system frequencies change more than 20%, appears to be the most interesting problem. For part A, if the move duration is longer than N cycles, the vibration caused by the initial acceleration should die out before the deceleration portion of the move begins. (I am assuming that a smooth trapezoidal velocity move is being used for these long moves, not a step command which would excite too much vibration.) Then only the deceleration is creating the residual vibrations. If the desired endpoint of the move is known, you should be able to shape just the deceleration for the system frequency at that position. This eliminates the need for an input shaper that takes into account the changing frequencies, either adaptively or by choosing a middle frequency.

One problem is choosing N, the number of cycles at which the initial acceleration dies out. This number depends on many factors, such as system damping and the percent change in system frequency. It is also possible that the deceleration does not excite residual vibration of significant amplitude. If the settling band is wide enough, input shaping might not be needed, if the smooth trapezoidal velocity profile is being used. Also, if the feedback controller has very good performance, it should be able to follow the trajectory closely with very little residual vibration. This was seen when the GSLQG controller was tested on trapezoidal velocity trajectories in Chapter 4.

For systems of type III.B, the vibrations from the acceleration of the arm up to speed and the vibrations from the deceleration will interact. If the move is timed carefully, the vibrations could be exactly out of phase and cancel each other. However, usually the vibrations add together and create vibrations of a higher amplitude than the deceleration alone would create. This seems like an ideal opportunity to use input shaping to get rid of the residual vibrations. There are many possible options. Simple shapers for one frequency, end or middle, are the easiest to implement and calculate. The initial acceleration could be shaped for the beginning frequency and the deceleration could be shaped for the end frequency. This is possible because there is a constant velocity section in between the accelerations, which allows the shapers to be switched smoothly. An adaptive shaping method that keeps switching between different input shapers as the frequency changes is also possible, though much more complex to implement.

5.3.2 Test Protocol

The purpose of this chapter is to investigate the workspace of the SRMS. Its frequencies shift as the geometry of the arm changes. As the arm moves from an elbow angle of 0 degrees to an elbow angle of 90 degrees, the frequencies change 25%. This indicates that the SRMS is a sensitive system, type III. In order to test my hypothesis, two different systems should be tested. One that models the situation in III.A (sensitive system, long duration move, and large frequency shift) and one that models the situation in III.B (sensitive system, short duration move, and large frequency shift). The problem is finding a system that will change frequency by at least 20% during a short duration move, thus representing III.B. The SRMS is a system of type III.A and the two-link model could be modified to represent III.B. By classifying the different systems into various sensitivities and move durations, the general problems involved with changing geometry systems should be decomposed into parts small enough to answer.

The long duration moves necessary to test III.A are done using trapezoidal velocity profiles. Unshaped runs of different move durations will show how much residual vibration there is after each move as a function of move duration. The test axes are sensitivity of the system and move duration. Another possible axis is system damping; however, the models represent the SRMS and changing the damping would change the system completely and the results would not necessarily translate to the SRMS. The goal of this thesis is to generate results that improve the base of knowledge about the SRMS, so damping was not chosen to be a major test matrix axis. The system damping sub-problem was investigated by running a small series of tests on the DRS model to see how changing the motor friction levels changed the system damping and response. These results are presented in section 5.5.6.

Figure 5.3 shows how the trapezoidal trajectories were calculated. t_1 is the length of acceleration. t_2 is the time at the end of the constant velocity section. *tce*, the end of the command, is equal to t_1+t_2, since the deceleration also takes t_1 seconds. There are two equations and five unknowns, a, d, v, t_1, t_2. d, the angle traveled, and a, the acceleration, are specified as a part of the test matrix. This leaves one variable to be chosen. In different tests, either v or t_2 was chosen and the others were calculated from equations shown in Equation 5.1.

$$\begin{aligned} v &= at_1 \\ d &= 0.5vt_1 + v(t_2 - t_1) + 0.5vt_1 = vt_2 = at_1t_2 \end{aligned} \tag{5.1}$$

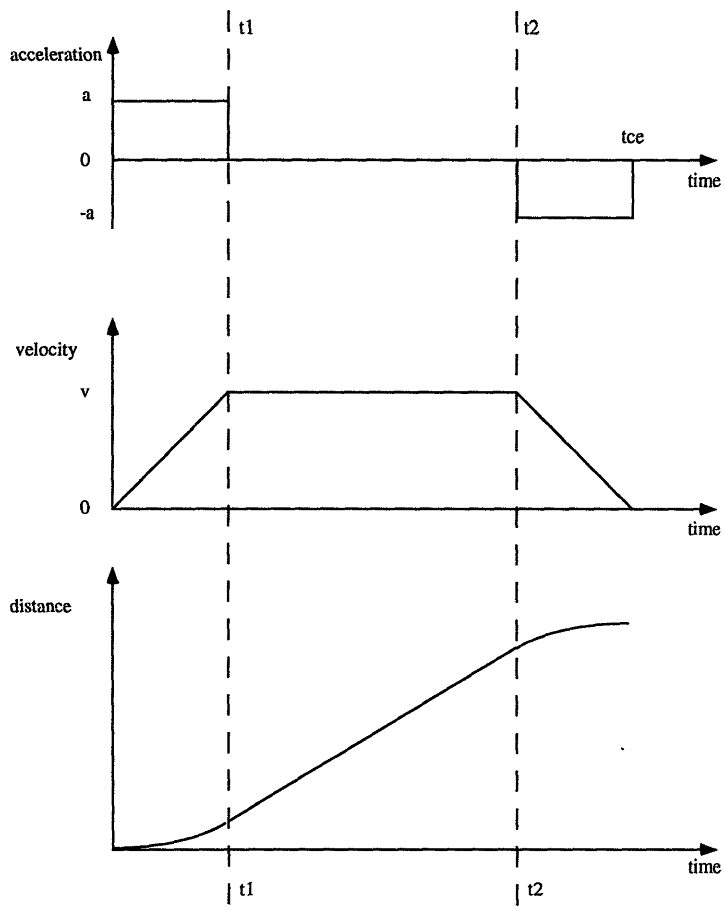


Figure 5.3: Trapezoidal velocity profile

The two-link model, the FLEX model, and the DRS simulation were run using the same variables. The velocity was varied over a range of values in the two-link and FLEX tests. For the DRS simulations, the velocity was chosen to be the maximum allowable for each payload, as given in Table 5.2. The acceleration, move distance, and payload were varied over the SRMS workspace. The payloads were chosen to be the unloaded case, the 7500 lb. payload used in the FLEX program, and a 32000 lb. cylinder payload.

5.4 Verifying and Testing Simple Models

5.4.1 Procedure

In order to verify that the two-link model's behavior represents that of the SRMS, it was compared to the $FLEX_2$ model. Trapezoidal velocity profiles were generated for the desired accelerations, velocities, and move distances. The move consisted of the elbow pitch joint moving an angular distance of $\Delta\theta_{elb}$=90 and 120 degrees, while the shoulder pitch was commanded to remain constant. The initial angles were $\theta_{sh}=0$ and $\theta_{elb}=0$. The accelerations ranged from 0.001 to 0.008 rad/s^2. The duration of the move was varied over a range that depended on the distance traveled. The appropriate performance metric is the time to settle after the command has finished, *tsc=ts-tce*, because the moves are all of different duration, which makes it hard to compare settling time. The settling times were calculated for a settling radius of 0.5 inches around the desired final tip position.

For the comparison to be valid, the $FLEX_2$ model was modified to take out the integrator in the rate controller. The new modified $FLEX_2$ model is called the $FLEX_{2m}$. Without the integrator, the rate controller settles much more quickly to a final position, though the position is not the desired one. The settling time for the $FLEX_{2m}$ is now defined to be the time taken to settle around its final position, not the desired position.

For every acceleration, velocity, and distance, there is a certain minimum move time when using a trapezoidal velocity profile. For D=-90 degrees and a=0.001 rad/s^2, the minimum move duration is 80 seconds. This means that there are already eight cycles of vibration (f~.10 Hz) for the shortest move. The percent frequency change is 35% and 24% respectively, in these two cases. For the two-link model with the 7500 lb. payload, the system must move more than 75 degrees to get over a 20% change in frequency. Therefore, this limits the minimum move length, and the hypothesis III.B cannot be checked by these models unless the torque limits are ignored and the acceleration increased.

5.4.2 Verification of the Two-link Model

Figure 5.4 shows two representative step responses from the $FLEX_{2m}$ simulation and the two-link model. The $FLEX_{2m}$ SRMS rate controller is the solid line, the two-link is the dashed line. The two models do not have the same damping ratio or follow the command in the same way, but the frequencies match and both responses settle at about the same time.

One of the main differences between the two models is the rise time. The $FLEX_{2m}$ response follows the velocity command very closely initially because a torque is being applied until the command goes to zero. Then the links, which have been bent by the acceleration, straighten. This causes the joint velocity to change directions. This behavior is shown in Figure 5.5, which is a closer look at Figure 5.4. The two-link model does not have torsional springs and thus no wind-up in the links, so its initial behavior differs from the $FLEX_{2m}$ response.

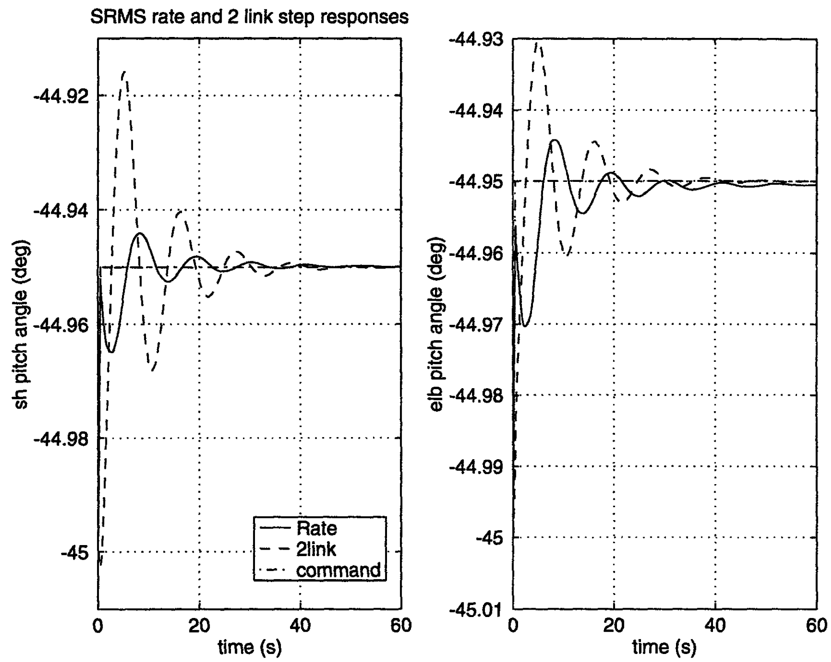


Figure 5.4: $FLEX_{2m}$ SRMS vs. 2LM step response

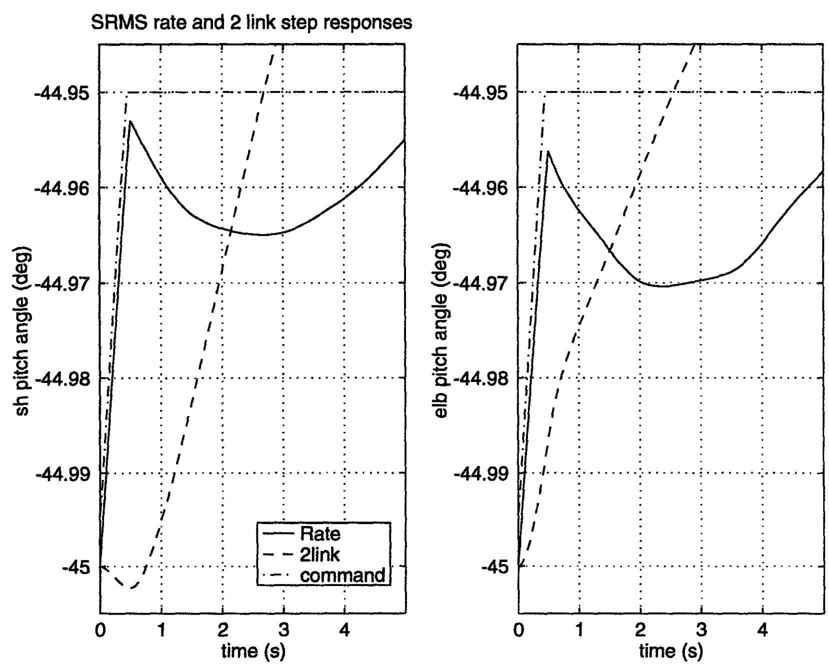


Figure 5.5: Detail of $FLEX_{2m}$ and 2LM step response

Table 5.3 shows the results for the two-link model and the $FLEX_{2m}$ rate controller model for a low acceleration and a range of velocities. For this acceleration, the two-link model unshaped response is settling before the command is over. The residual vibrations have amplitude less than half an inch, so the responses are settling during the initial rise. The $FLEX_{2m}$ unshaped response is settling a short time after the command is done. The two different models do not have exactly the same system response, but the results are similar. Decreasing the velocity while increasing move duration is not affecting the settling time except in the last two cases of the 2LM. For D=90, v=0.0132 rad/s, there is an overshoot of the desired x tip position. The overshoot has an amplitude of 0.84 inches, so it is just larger than the settling radius. The elbow angle overshoot can be seen in Figure 5.8, which shows the two-link model responses for two different velocities and D=90. The initial overshoot is increasing with increasing command length. This causes the settling time to be after the command is done.

D=90 degrees			2LM	2LM	SRMS rate	SRMS rate
amax	vmax	tce	unshaped	neg. EI	unshaped	ZVD
(rad/s^2)	(rad/s)	(sec)	(sec)	(sec)	(sec)	(sec)
0.001	0.0394	79.4	-2.8	2.65	1.2	5.1
0.001	0.0316	81.6	-2.8	2.60	1.2	5.1
0.001	0.0262	86.2	-3.0	2.65	1.2	5.1
0.001	0.0226	92.6	-2.8	2.65	1.2	5.1
0.001	0.0198	99.8	-2.8	2.60	1.0	
0.001	0.0176	107.6	-3.0	2.60	1.0	5.1
0.001	0.0158	115.8	-3.0	2.65	1.2	
0.001	0.0144	124.4	0.0	2.65	1.4	5.1
0.001	0.0132	133.2	0.2	2.65		

Table 5.3: Settling times for 2LM and $FLEX_{2m}$, D=90, a=0.001

When input shaping was added to the low acceleration moves, it only added a delay to the system, as shown in Table 5.3. Two different types of input shapers were tried, a ZVD and a negative EI input shaper. The frequencies were chosen to be the frequencies of the system at the end of the move. Because the unshaped responses were settling during the initial rise or right after the end of the command, there was not much that input shaping could do. The low acceleration and smooth trapezoidal velocity profile gets rid of all of the residual vibration. There is not need for input shaping at these values.

Figure 5.6 shows the time history of the elbow angle for the two-link and SRMS rate controller for a=0.001 rad/s^2, v=0.042 rad/s, D=-120 degrees. The $FLEX_{2m}$ is not following the command during the deceleration, and once the command ends, settles quickly, though to the wrong position. If the integrator was still in place, the SRMS rate controller would settle to the correct position, but would take an extra 50 seconds. The amplitudes of the residual vibration for both models are both about 0.24 inches. The second subplot shows that the amplitude of residual vibration is low for both models. The two-link has better performance at trajectory following than the SRMS rate controller, since it has position and velocity feedback

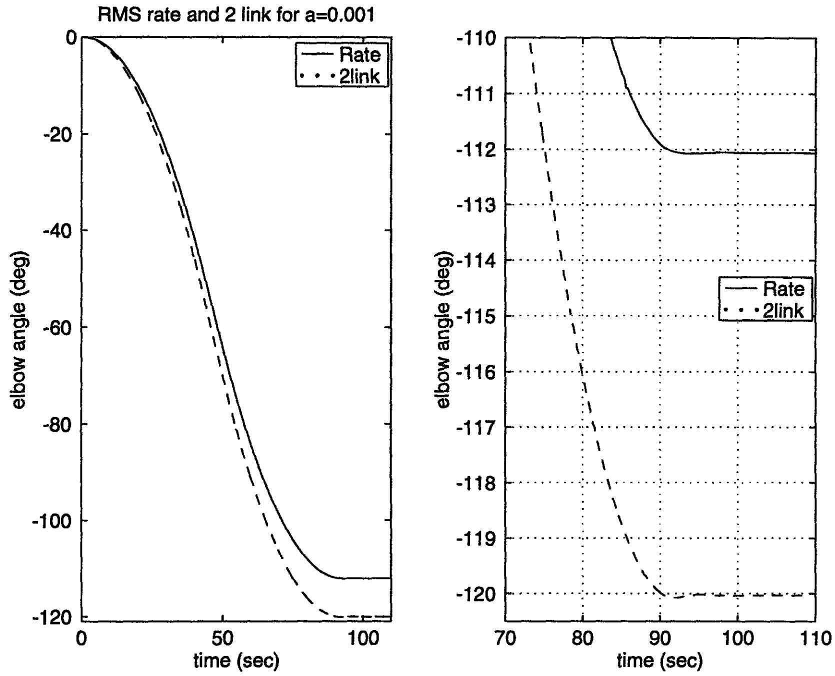


Figure 5.6: Time history for 2LM and $FLEX_{2m}$, D=120, a=0.001

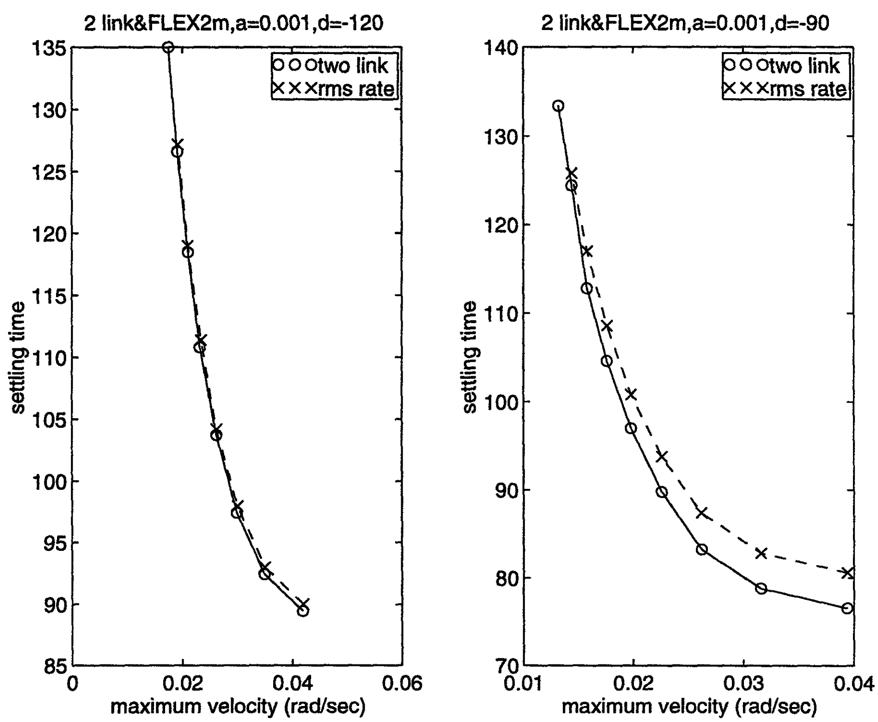


Figure 5.7: Settling times for 2LM and $FLEX_{2m}$, D=90 and 120, a=0.001

Figure 5.7 shows a comparison of settling times for the two-link model and the $FLEX_{2m}$ SRMS rate controller model. When looking at the individual time histories, the two-link model and the $FLEX_{2m}$ follow the same trend, but the

FLEX simulation does not follow the command as well as the two-link model because of saturation and friction. However, the two models do perform roughly the same so results generated on the simpler and quicker two-link model should translate to the more complicated $FLEX_2$ model and the DRS.

5.4.3 Two-link Model Results

Once the two-link model was verified by testing it with the $FLEX_2$ model, more tests with trapezoidal profiles were run. Figure 5.8 shows the 2LM residual vibration and command for two different velocities and a 90 degree move. The quicker response has a near-triangular velocity profile, v=0.039 rad/s which lasts 79.4 seconds. The slower response has a command time of 133 seconds and a velocity one-third of the previous velocity. There is not very much vibration in either response; however, the lower velocity has slightly more residual vibration, enough to make it settle just after the command is over. Increasing the velocity is having very little effect on the amount of residual vibration. One possible reason is that the acceleration is too low. The acceleration was varied in the next set of tests. Because there was not much residual vibration in the low acceleration moves, input shaping was only added to the higher acceleration moves.

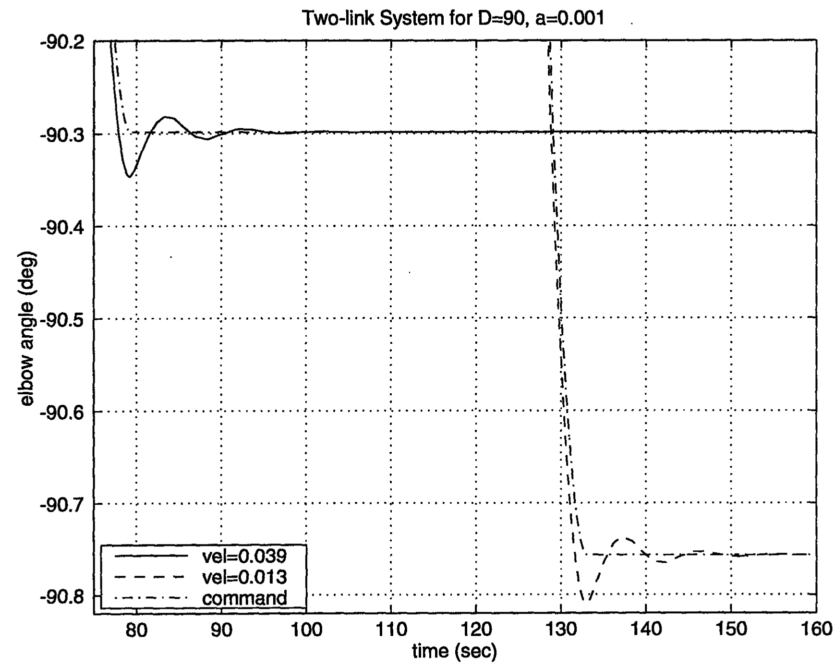


Figure 5.8: 2LM large slew move for two velocities, D=90 degrees, a=0.001 rad/s^2

Table 5.4 shows the results for an acceleration of 0.004 rad/s^2, where the move distance equals 90 degrees. The unshaped results settle after the command has ended. The negative EI input shaper does improve performance here; it saves several seconds. However, compared to the length of the command, not much time is saved at all, there is only a 4% time savings. The negative EI input shaper lasts 6.09 seconds, so it is getting rid of all the vibration and allowing the response to settle before the shaped command has ended. The ZVD shaper is doing the same thing, but its duration is 8.90 seconds and its longer time delay means that the ZVD responses settle after the negative EI do.

D=90			2LM settling times (ts-tce)		
amax	vmax	tce	unshaped	ZVD	neg. EI
(rad/s^2)	(rad/s)	(sec)	(sec)	(sec)	(sec)
0.004	0.0786	39.65	5.70	6.05	4.80
0.004	0.0524	43.10	6.00	6.00	4.80
0.004	0.0392	49.80	5.60	6.00	4.80
0.004	0.0314	57.85	5.25	6.00	4.75
0.004	0.0262	66.55	5.55	6.00	4.75
0.004	0.0224	75.60	8.90	6.00	4.75
0.004	0.0196	84.90	9.40	6.05	4.80
0.004	0.0174	94.35	9.70	6.05	4.85
0.004	0.0158	103.95	9.65	6.10	4.90

Table 5.4: Settling times for 2LM, D=90, a=0.004

When the acceleration is increased to a=0.008 rad/s^2, the level of vibration increases, shown in Table 5.5. The unshaped responses take longer to settle, as do the shaped responses. The lengths of the input shapers are the same, but the additional vibration causes the shaped responses to settle slightly after the lower acceleration shaped responses.

D=90			2LM settling times (ts-tce)		
amax	vmax	tce	unshaped	ZVD	neg. EI
(rad/s^2)	(rad/s)	(sec)	(sec)	(sec)	(sec)
0.008	0.1120	28.00	10.60	6.75	7.70
0.008	0.0784	29.80	6.35	6.75	5.30
0.008	0.0524	36.55	9.80	6.75	7.75
0.008	0.0392	44.90	13.70	6.75	8.50
0.008	0.0316	53.95	13.25	6.75	8.75
0.008	0.0260	63.25	11.00	6.80	8.90
0.008	0.0224	72.80	11.05	6.85	8.90

Table 5.5: Settling times for 2LM, D=90, a=0.008

The ZVD shaper is doing a better job at minimizing residual vibration for this acceleration than the negative EI shaper. Both shapers have the same insensitivity, but the EI allows 5% of the residual vibration to remain rather than trying to get rid of all the vibration. These results indicate that the frequencies are shifting slightly with acceleration. The EI shaper did better for a=0.004 rad/s^2, but the ZVD shaper did better for a=0.008 rad/s^2. There is no good explanation for why one shaper did better at one acceleration and another shaper did better at a different acceleration.

The faster acceleration runs are definitely producing more vibration. The response no longer settles before the command ends, and the runs end much more quickly. Table 5.6 shows a comparison of the results for different accelerations and the same velocity. Not only are the unshaped settling times increasing, but the shaped settling times are increasing as well, even though the duration of the input shaper stays constant.

D=90			2LM settling times (ts-tce)		
amax	vmax	tce	unshaped	ZVD	neg. EI
(rad/s^2)	(rad/s)	(sec)	(sec)	(sec)	(sec)
0.001	0.0394	79.4	-2.8		2.65
0.002	0.0393	59.65	1.45		3.75
0.004	0.0392	49.80	5.60	6.00	4.80
0.008	0.0392	44.90	13.70	6.75	8.50
tis			0	8.90	6.09

Table 5.6: Settling times for 2LM, varying accelerations

Figure 5.9 plots the *tsc* vs. the velocity from Table 5.4 and Table 5.5. The negative EI shaper has a constant settling time, regardless of velocity and acceleration. The ZVD gives a constant settling time for the lower acceleration, but follows the unshaped performance more closely for the higher acceleration. Part of the changes may be due to the metric used. Settling time to a certain radius is not a very robust metric. If the system barely misses the settling radius, it is not taken into account and the response does not settle until the next cycle is over. This may explain the irregularity in the curves, especially the unshaped responses. The input shaper time delay means that the responses settle before the shaped command is through. Since there is no vibration after the end of the move, the settling time is equivalent to the rise time.

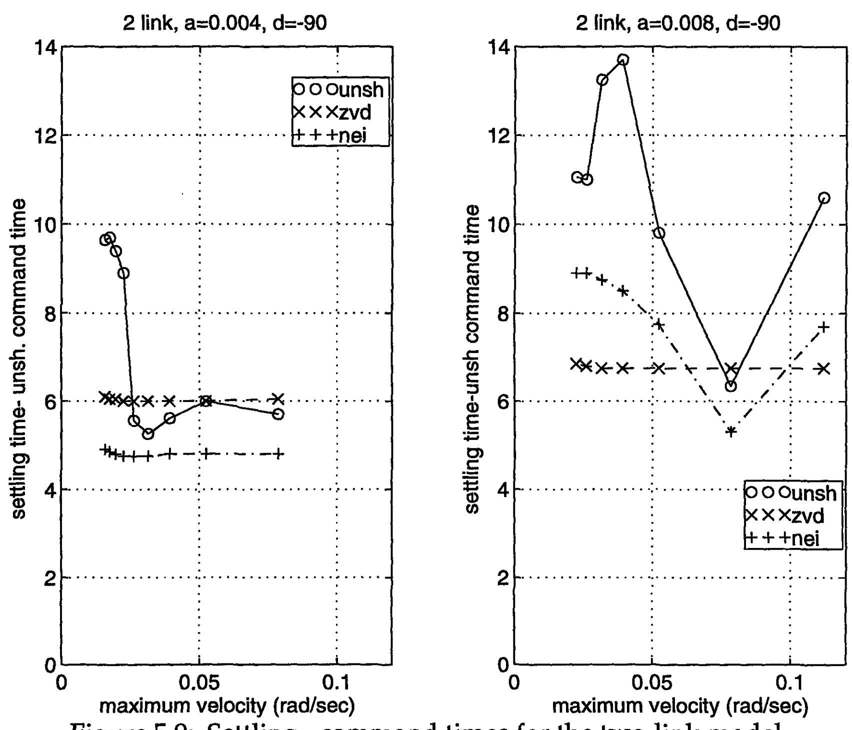


Figure 5.9: Settling - command times for the two-link model

From these results, I draw the conclusion that higher acceleration moves do excite more vibration. The two types of input shapers implemented reduced the residual vibrations in the higher acceleration moves. However, the shaped responses settle before the shaped command has finished, so a shorter input shaper might still get rid of all of the vibration and have less delay.

5.4.4 *Conclusions*

A simple two-link model was developed and tested. It used the SRMS parameters and was tuned to represent the SRMS first frequency and damping ratio. When the performance was compared with the $FLEX_2$ simulation, the agreement was good. This model is much simpler and thus runs faster than the $FLEX_2$ simulation which must calculate mode shapes and torsional effects.

From the initial investigation of the two-link and $FLEX_2$ models, input shaping is only needed for a long move if there is a very tight settling criterion. If you only need to settle to within a one inch radius or greater, then the unshaped moves settle on the way up to the final desired position. The acceleration also has a large effect. If the acceleration is increased, the move ends more quickly with more residual vibration. However, the residual vibration has an amplitude of less than two inches even for the higher accelerations. Trapezoidal velocity profiles excite far less vibration than step commands.

For the very small radius, the ZVD and negative EI shapers get rid of vibration. They have an associated time delay that is very close to how long the unshaped case takes to settle. Therefore, we are not getting much time savings at all. As the acceleration is increased, the model is having more trouble keeping up with the trapezoidal profile, and thus has more vibration at the end of the move. The next section will examine more closely the problem of how acceleration, move distance, move duration, and input shaping interact for trapezoidal velocity profiles.

5.5 DRS Results

The DRS model was initially just going to be used to verify the FLEX simulation's performance. However, it proved to be simpler to understand than expected. Therefore, a series of tests were run using it, since it has been verified by NASA. It should be very accurate so that recommendations gathered from its results can be transferred to the SRMS without modification or the need for much more further verification.

5.5.1 *Simulations Run*

Runs were done for maximum velocities, three different payloads, a range of accelerations, and a range of distances. The range of accelerations varies from payload to payload, due to the differing inertias and the maximum motor torque. The unloaded payload has a maximum velocity of 3.21 deg/s and a maximum acceleration approximately of 0.05 rad/s^2. The 7500 lb. payload case has a maximum velocity of 0.51 deg/s and a maximum acceleration of roughly 0.005

rad/s^2. The 32000 lb. payload has a maximum velocity of 0.321 deg/s and a maximum acceleration of 0.001 rad/s^2. These maximum velocities were specified by NASA. The maximum acceleration were determined by looking at the simulation results that follow in this section. Table 5.7 shows the test matrix used in this section.

Axes	P=0	P=7500	P=32000
Distance	15,45,90	15,45,90,135	15,45,90
Acceleration (rad/s2)	0.002-0.064	0.001-0.016	0.0002-0.008
Elbow velocity (deg/s)	3.21	0.51	0.32
Input Shapers	ZV, negative ZV, ZVD, negative ZVD, EI, negative EI, ZVb		

Table 5.7: DRS test matrix

The moves were chosen to be joint space moves instead of Cartesian space moves to make it easier to define and calculate the moves and angles. The distance of the move was defined to be the angular distance the elbow pitch joint moved. The shoulder pitch angle moved only one-quarter of the distance the elbow moved. I chose to move both of the joints at once to represent realistic behavior of the arm. In space, it is likely that both joints would be moving, not just the elbow joint. The distances chosen were 15 degrees, 45 degrees, and 90 degrees. The initial angles were chosen to be θ_{sh}=0 and θ_{elb}=0. Figure 5.10 shows the initial and final arm positions for each of these moves.

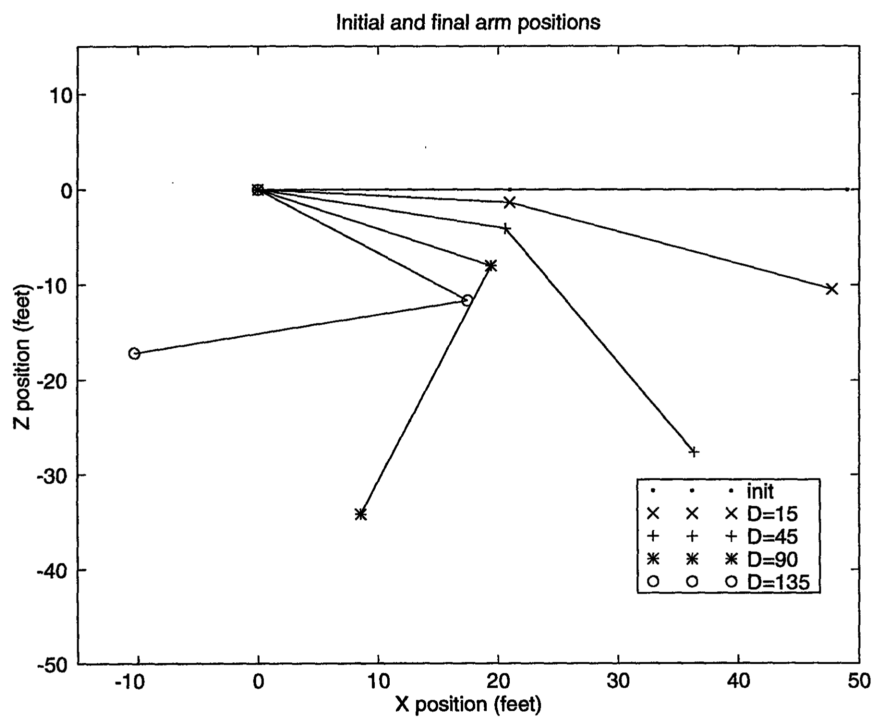


Figure 5.10: DRS arm positions for various move distances

The metric used to measure performance is settling time. Because all of the moves have different durations, the time to settle after the command has finished, *tsc*, will be used. *tsc* is defined to be the overall settling time, *ts*, minus the command time, *tce*, will be used again. However, this metric does not adequately capture the system's performance. If the system response just misses a settling radius, the *tsc* will not indicate this. Thus, a new settling time metric was developed. The system tip positions were fed into a MATLAB function where the frequency of the residual vibration was calculated. Then an exponential decay envelope was fitted to the residual vibration to get the damping ratio. Once the damping and frequencies were calculated, the envelope settling time could be calculated. This was defined to be the point at which the exponential envelope hit the settling radius of 0.5 inches. After much refinement, this function worked well and calculated the envelope settling time, *tse*, for any payload, tip position, or combination of several tip positions. This metric is more consistent than the regular settling time metric.

One of the problems with this metric occurs when the residual amplitude is much lower than the settling radius. Then the envelope settling time calculation may give a time that occurs before the move has started. To fix this problem, the envelope settling time was cut off at the end of the command. If the *tse* was less than the duration of the command, it was set equal to the length of the command. This will be seen in later tables where entire columns are equal to the length of the shaped command. In these cases, the regular settling time gives an accurate portrayal of how much earlier the response settled.

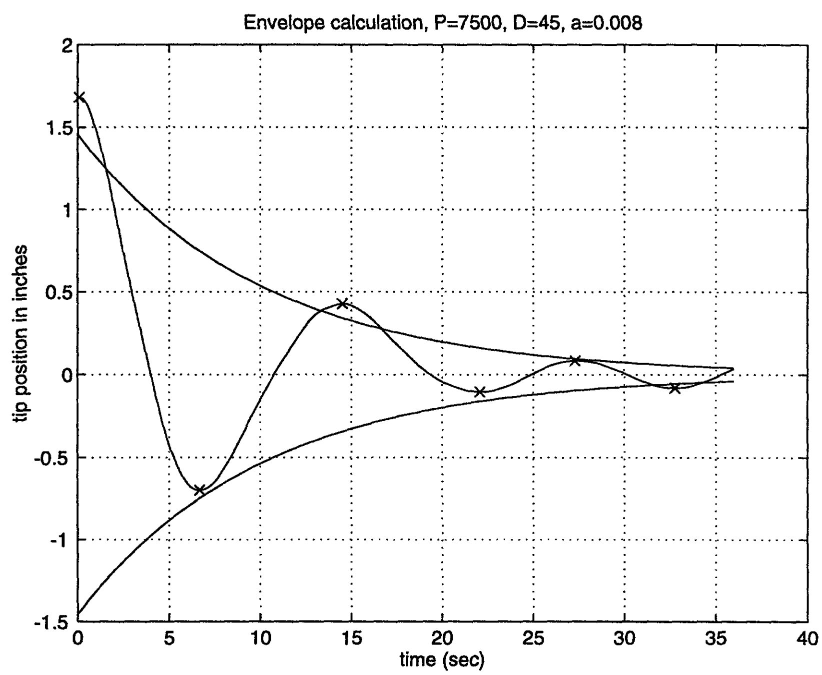


Figure 5.11: Calculation of exponential envelope

Another problem is that the envelope calculation is an automatic function run by MATLAB. The function fits the exponential curve to the responses as best it can. Usually it works very well and the exponential envelope contains all of the residual oscillations. However, sometimes the fit is not perfect, as shown in Figure 5.11. In theory the exponential envelope is the maximum and none of the response is outside of the envelope. Then the envelope settling time is always the same as or longer than the radius settling time. But since the function does not always find a perfect exponential fit, the system response can escape from the envelope. When this occurs, the regular settling time is longer than the envelope settling time. In Figure 5.11, the response goes outside the envelope just after the envelope has settled to 0.5 inches. Thus the regular settling time is longer than the envelope settling time. An exponential window or Fourier transform of the data could solve some of these problems but were not investigated.

Many different input shapers were calculated and implemented on the DRS runs. In the changing geometry problem, it is not clear which frequency should be chosen for the input shaper. The end frequency is a popular choice, as is the middle frequency. Both were tried here. The types of shapers used are as follows: ZV, negative ZV, ZVD, negative ZVD, EI, negative EI. These shapers have been described in Chapter 2. A modified ZV shaper was also implemented that shapes the initial acceleration for the start frequency and the deceleration for the end frequency; it will be called ZV_b.

The result tables in the following sections list settling times in columns. Each column represents a different input shaper. If the shaper name is followed by a *m*, the suffix means that shaping for the middle frequency rather than the end frequency was done. The *b* suffix indicates that two different shapers were used; the acceleration up to constant velocity was shaped for the beginning frequency and the deceleration was shaped for the end frequency. The *e* suffix means shaping for the end frequency. *tis* is the duration of the input shaper and is given in the last row of the table. *thelbf* is the final elbow angle. In the tables, the best or two best input shapers at each acceleration are shaded to make them easier to distinguish from the others.

The shaper frequencies were calculated several different ways and are shown in Table 5.8. Since the unshaped responses were simulated before the shaped responses, the envelope settling function were used to find the frequency of the residual vibration and damping ratio at each position. These values were used to create the first set of input shapers. These shapers have no suffix in the result tables. The second set of frequencies was found by giving the system a small velocity step of duration 0.25 seconds at the desired angle. Once again, the envelope function was used to find the system frequencies at that angle from the step response. The input shapers with suffix 2 were done using this second set of frequencies. These frequencies were quite different than the first frequencies used. When this was realized, the frequencies were recalculated with a velocity step of 1.25 seconds. These frequencies were much closer to the original frequencies. The input shapers with suffix 3 are done with the third set of frequencies. The problem with finding the frequencies was that the system is nonlinear, so it oscillates with changing periods of oscillations. The envelope function was used to find the frequencies, but it is not perfect, so the frequencies are not perfect either.

Payload (lb.)	Elbow Angle (degrees)	Frequency 1 (Hz)	Frequency 2 (Hz)	Frequency 3 (Hz)
0	0	0.478	0.460	0.459
0	15	0.384	0.449	0.412
0	45	0.376	0.485	0.447
0	90	0.592	0.588	0.582
7500	0	0.070	0.090	0.083
7500	15	0.071	0.091	0.076
7500	45	0.073	0.090	0.075
7500	90	0.090	0.105	0.098
32000	0	0.048	0.037	0.037
32000	15	0.037	0.040	0.036
32000	45	0.043	0.036	0.040
32000	90	0.048	0.039	0.047

Table 5.8: DRS frequencies

In the result tables in the following sections, certain cells are shaded. These cells contain the shaped or unshaped responses that had the fastest settling times. If the shaped responses settled after the shaped command was done, the envelope settling time, *tse*, was used to find the best response. If the shaped response settled before the shaped command was done, the regular settling time, *tsc*, was used because the *tse* would automatically be the length of the input shaper delay. So in each row, either the envelope or the regular settling time was picked to be best, not both.

5.5.2 General Trends in Unshaped Slews

The unshaped settling times for the midsize payload are shown in Figure 5.12. The unshaped responses are very similar for commanded acceleration, a, greater than 0.002 rad/s^2. Only at the lower accelerations does the amount of residual vibration change much at all. At a certain commanded acceleration, the motor saturates and the arm can no longer keep up with the commanded profile. At the end of the saturated move, the motor decelerates as fast as it can, but for a duration that depends on the specified acceleration. This phenomenon occurs because the DRS has a rate controller and not a position controller. So if the commanded acceleration is above the maximum motor acceleration, the response should decelerate from the constant velocity towards zero velocity. However, the deceleration is too quick for the arm to be able to follow the command, so the command will go to zero velocity before the response has reached zero velocity. This sudden ending of the command should cause residual vibration to increase, even when above saturation levels. The duration of the acceleration, t_1, does level off above certain accelerations. For example for P=0, t_1=1.17 for a=0.048 rad/s^2 but t_1=0.88 for a=0.064 rad/s^2. This explains why settling times do not increase much once the acceleration is above a certain level; there is not much difference in deceleration duration.

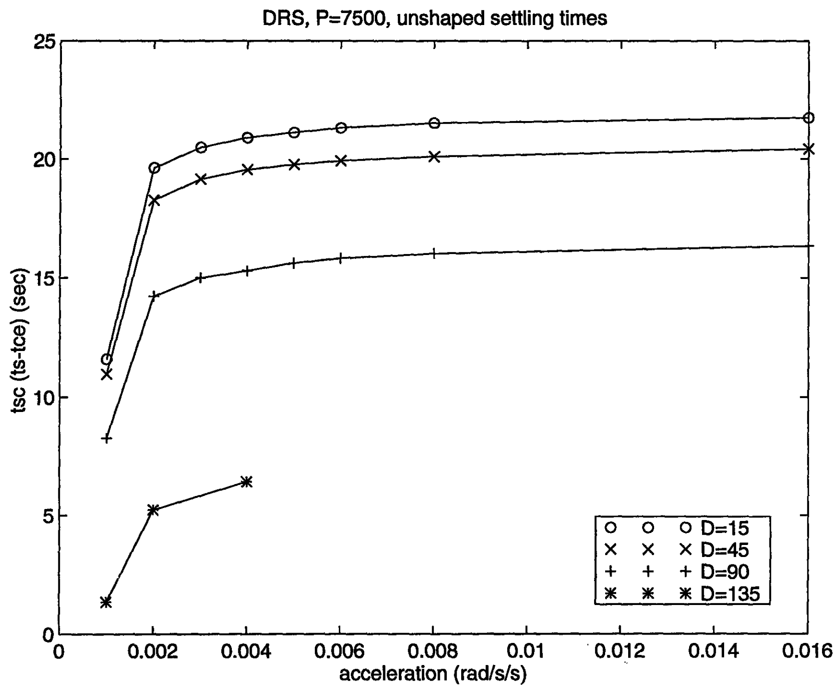


Figure 5.12: Unshaped settling times for P=7500

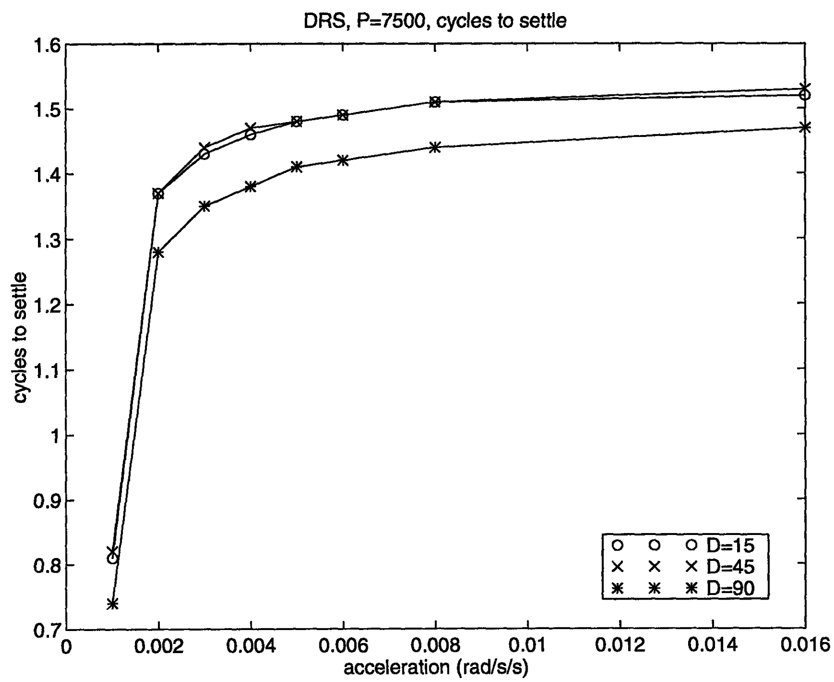


Figure 5.13: Cycles to settle for P=7500

The midsize payload's fundamental frequency varies from 0.07 Hz at an elbow angle of zero degrees to 0.1 Hz at 135 degrees. Thus one cycle of vibration will last between 10 to 14 seconds. As the move distances rise, the amount of vibration of the system actually does not change much, shown in Figure 5.13. For D=15 and D=45, the cycles to settle are almost equal. The number of cycles to settle decreases some for D=90, but not by much. For the D=135 case, there is much less vibration. One of the main reasons for this is that the moves last around 270 seconds. That means that there are 27 cycles for the initial vibration caused by the acceleration to die out. The deceleration is also at an arm position where it has very little inertia compared to its initial arm position, as shown in Figure 5.10. Consequently, it has less residual vibration.

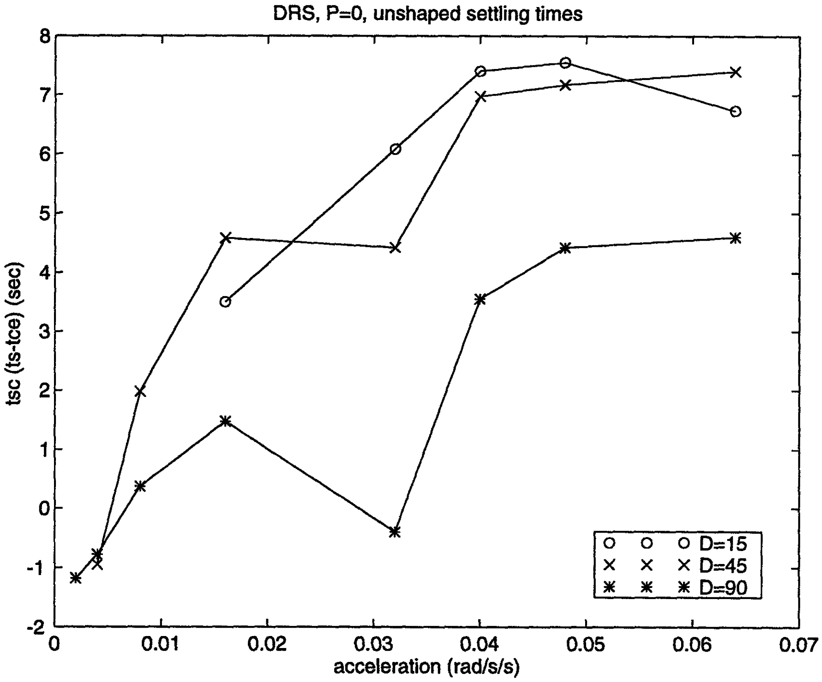


Figure 5.14: Unshaped settling times for P=0

For the unloaded payload case, the amplitude of vibration increases as acceleration grows, as shown in Figure 5.14. The motor is saturating at an acceleration around 0.05 rad/s2. The D=15 case could not be run for accelerations below a=0.01 rad/s2 because the move distance is short enough that it takes longer to reach the maximum velocity and then decelerate than it does to move the required distance. The interesting dip in the settling time curves is characteristic of trapezoidal velocity profiles. At certain lengths of accelerations, the acceleration steps act like a ZV input shapers. If the initial acceleration lasts an even multiple of the system's period, then the acceleration can be considered two impulses of a ZV zero damping input shaper convolved with a step command. (A zero damping ZV shaper has two pulses of amplitude 0.5 and separated by half a period.) If the second half of the initial acceleration is exactly out of phase with the first half, then the second acceleration oscillations cancel the first acceleration oscillations exactly. Or if the deceleration is exactly out of phase with the initial acceleration, the oscillations can also cancel each other out. This behavior occurs at the same point for D=45 and D=90. For D=15, the vibrations appear to be canceling around a=0.064 rad/s2. Christian has a more complete explanation of the behavior of trapezoidal velocity profiles. [6]

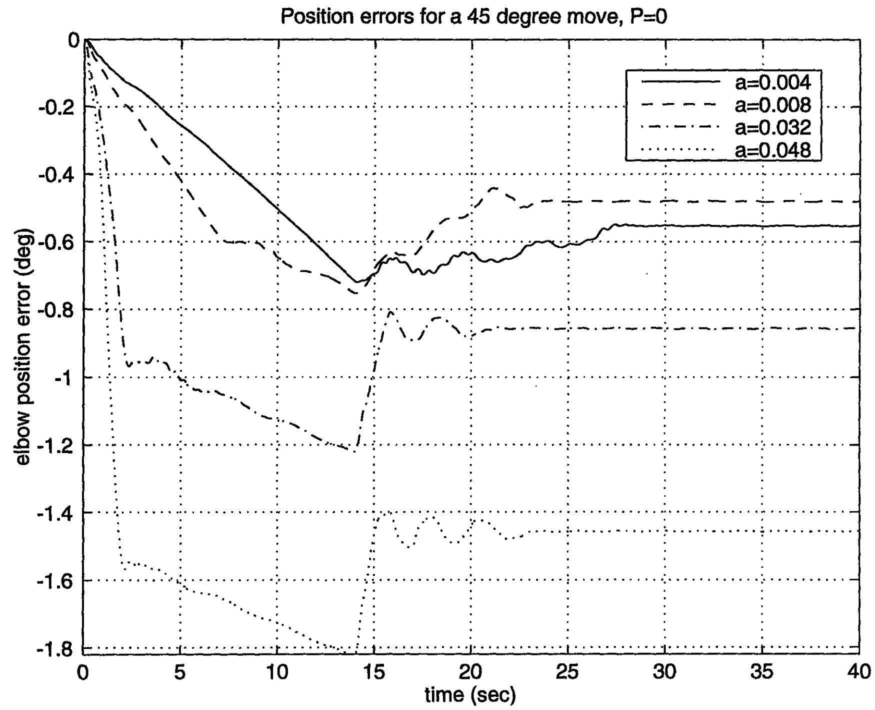


Figure 5.15: Position errors for varying accelerations, P=0, D=45

Figure 5.15 shows the different position errors for several accelerations. The first part of the curve is the initial acceleration. No vibration can be seen during this part of the move. The next section is the constant velocity section. Different amounts of vibration can be seen here. The third part of the move is the deceleration and the fourth part is after the command is done, where only the residual vibrations are left. For a=0.004 rad/s^2, the move is a triangular velocity profile, so there is no constant velocity part. For this acceleration, the vibrations can be seen during the deceleration. The duration of the acceleration, t_1, is only a function of velocity and acceleration and not the distance traveled, as illustrated in Equation 5.2. t_2-t_1 is the length of the constant velocity section of the move.

$$t_1 = \frac{v}{a} \qquad t_2 = \frac{d}{v} \tag{5.2}$$

The maximum acceleration is even less for the large payload case due to the large payload inertia, as shown in Figure 5.16. The interesting effect here is that the D=15 and D=90 cases follow each other very closely, while the D=45 case has the dip caused by the acceleration and deceleration canceling.

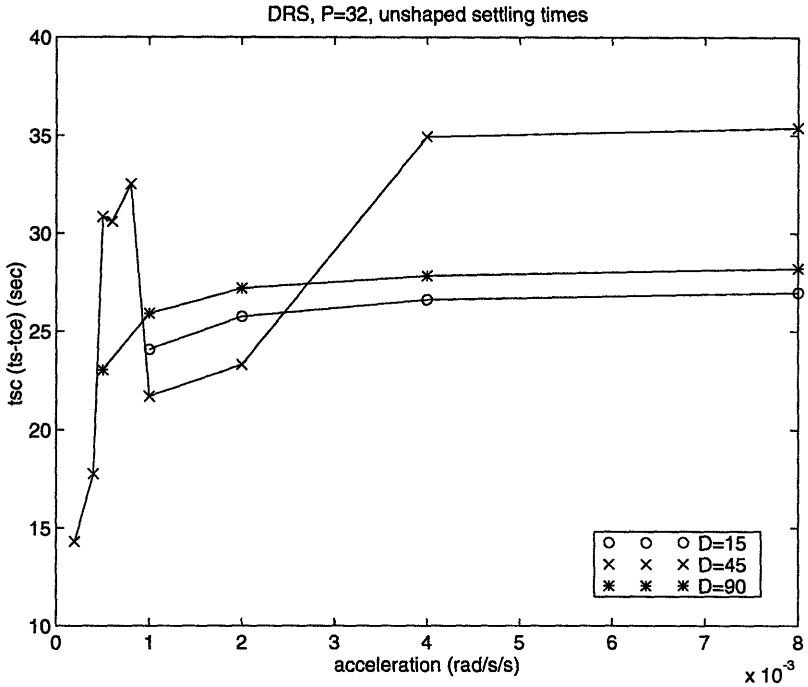


Figure 5.16: Unshaped settling times for P=32000

Certain trends can be seen in the unshaped data. As acceleration increases, the amount of residual vibration also increases. This is true for all of the payloads and distances, except when the acceleration and deceleration are timed to cancel each other. However, at some point the amount of residual vibration levels off and increasing the acceleration does not increase the amplitudes of oscillations. This phenomenon occurs because the constant velocity of the trapezoidal profile is fixed according to the limits of each payload. As the acceleration increases, at some point raising the acceleration only decreases the move duration by less than a second.

5.5.3 Midsize Payload Results

For an average-sized payload of 7500 lb., the best shapers are the ZVD and the negative ZVD. The ZVD's envelope settling times are equal to the length of the input shaper, which means that the response is settling before the end of the shaped command, so we have to look at the regular settling time to see when it actually settles. The shapers are reducing the vibration enough so that the response settles while rising to the final position. The performance is improved by the use of input shaping.

For the P=7500 and D=45 case, the ZVD shaper family is doing the best job. The results are given in Table 5.9. The negative ZVD and the ZVD get rid of the most vibration. However, not much time is saved since the ZVD shaper delay lasts one period and the vibration is dying out in approximately two periods. At most we save 12 seconds on a 92 second move, an 11% time savings. Interestingly, the negative EI does well for a=0.002, but not for a=0.004 or 0.008.

The ZVD and EI responses are settling before the shaper is done. Evidently the additional robustness helps here.

Regular settling times, tsc (sec) Payload=7500, D=45

accel	tce	unsh	ZV	ZV2	ZV3	neg ZV	neg ZV3	ZVD	ZVD2m	ZVD3	neg ZVD	neg ZVD2	neg ZVD3	EI	neg EI	neg EIm
0.001	97.13	10.97														
0.002	92.68	18.27	17.00	16.90	18.20	15.62	16.65	8.77	14.65	17.42	6.47	13.10	15.85	9.15	7.72	13.12
0.003	91.20	19.15		17.35	18.70		17.30	9.30	15.38	18.30	6.88	13.70	16.80			
0.004	90.45	19.55	18.55	17.62	19.02	20.70	17.25	9.65	15.77	18.75	12.40	14.00	17.15	10.12	12.05	20.00
0.005	90.00	19.77		17.77	19.20		17.25	9.87	16.02	19.02	14.10	19.97	17.40			
0.006	89.70	19.92		17.90	19.35		17.30	10.02	16.20	19.22	14.58	19.97	17.40			
0.008	89.35	20.10	18.87	18.02	19.47	22.45	17.37	10.20	16.35	19.42	14.77	20.80	23.67	10.72	13.72	21.35
0.016	88.78	20.42														
tis		0.00	6.96	5.55	6.67	3.22	3.88	13.92	11.09	13.33	9.87	7.54	9.08	13.70	9.97	7.60

Envelope settling times, tse (sec) Payload=7500, D=45

accel	tce	unsh	ZV	ZVm	ZV3	neg ZV	neg ZV3	ZVD	ZVD2m	ZVD3	neg ZVD	neg ZVD2	neg ZVD3	EI	neg EI	neg EIm
0.001	97.13	10.16														
0.002	92.68	17.03	14.14	15.47	16.29	16.17	16.57	13.92	11.09	16.91	9.87	11.89	14.98	13.70	9.97	14.45
0.003	91.20	17.97		17.83	18.22		17.78	13.92	16.46	17.74	9.87	15.29	17.87			
0.004	90.45	18.37	14.90	18.54	18.81	19.72	19.28	13.92	16.43	20.36	9.87	16.33	18.85	13.70	9.97	16.81
0.005	90.00	18.54		19.03	19.16		20.02	13.92	17.00	20.63	12.26	18.07	20.66			
0.006	89.70	18.72		19.15	19.32		18.82	13.92	16.96	20.76	13.59	16.95	20.91			
0.008	89.35	17.96	15.34	19.19	19.72	20.06	19.12	13.92	18.13	20.92	13.62	17.30	21.46	13.70	14.15	17.89
0.016	88.78	18.27														
tis		0.00	6.96	5.55	6.67	3.22	3.88	13.92	11.09	13.33	9.87	7.54	9.08	13.70	9.97	7.60

Table 5.9: Settling times for P=7500, D=45

Regular settling times, tsc (sec) Payload=7500

thelbf	accel	tce	unsh	ZV	ZVm	ZV3	neg ZV	neg ZV3	ZVD	ZVD2	ZVD3	neg ZVD	neg ZVD3	neg EI
15	0.0010	38.30	11.60	10.50										
	0.0020	33.85	19.62	18.27	17.50	18.75	16.23	17.23	9.30		17.52	6.62	17.23	7.73
	0.0030	32.38	20.48		17.92	19.17		17.62	9.77	15.27	18.33	6.95	17.62	
	0.0040	31.63	20.90	19.62	18.20	19.47	22.30	17.67	10.15		18.67	7.07	17.67	12.15
	0.0050	31.18	21.12		18.32	19.65		17.85	10.35	15.82	18.93	14.25	17.85	
	0.0060	30.88	21.32		18.42	19.77		17.92	10.52	15.97	19.07	14.55	17.92	
	0.0080	30.53	21.52	20.27	18.57	19.93	23.68	18.02	10.68		19.27	14.85	18.02	14.22
	0.0160	29.95	21.75											
	tis		0.00	7.14	5.52	6.60	3.20	3.84	14.29	10.98	13.19	10.11	8.98	10.21

Envelope settling times, tse (sec)

thelbf	accel	tce	unsh	ZV	ZVm	ZV3	neg ZV	neg ZV3	ZVD	ZVD2	ZVD3	neg ZVD	neg ZVD3	neg EI
15	0.0010	38.30	13.43											
	0.0020	33.85	16.24	14.72	16.83	17.46	16.77	17.86	14.29		20.01	10.11	15.97	10.21
	0.0030	32.38	19.17		18.74	19.44		19.58	14.29	17.21	20.89	10.11	19.84	
	0.0040	31.63	19.70	16.36	19.49	19.98	20.12	20.79	14.29		21.27	10.11	20.60	10.21
	0.0050	31.18	19.90		19.82	20.31		21.22	14.29	17.84	21.42	11.47	20.92	
	0.0060	30.88	20.13		20.00	20.53		21.50	14.29	13.19	21.61	12.21	21.05	
	0.0080	30.53	20.34	17.22	20.34	20.79	21.28		14.29		21.82	12.84	21.30	14.19
	0.0160	29.95	20.54											
	tis		0.00	7.14	5.52	6.60	3.20	3.84	14.29	10.98	13.19	10.11	8.98	10.21

Table 5.10: Settling times for P=7500, D=15

When P=7500 and D=15, the ZVD shaper family does the best as shown in Table 5.10. In fact the results here mirror the results from the D=45 case. The ZVD and negative ZVD all trade off being the best; the negative ZVD does the best for lower accelerations, but the ZVD does the best for higher accelerations. The time savings is about 12 seconds for a 30 second command and 20 second *tsc*; a time savings of 24%. The ZV shapers are not doing well at all. They do slightly better than the unshaped case, but only save a second or two. The negative ZV is doing worse than the ZV. The shapers that use the first frequency, from the unshaped trapezoidal responses, do better than the shapers of the same type that use frequencies 2 or 3.

For the P=7500, D=90 case, the ZVD family is still giving the best performance. The negative ZVD gives the highest increase in performance by saving about 10 seconds for all accelerations. Table 5.11 gives the results. The ZV_b shaper, shaping for both the beginning and end frequency, was tried but it did worse than the regular ZV shaper. This is a longer move and at the end position there is a dominant x mode; the z mode settles on the way up. These input shapers are only designed for one mode, so might do better if the tip is only vibrating in one direction. This might explain the problems at D=45. At the end of that move, the tip is vibrating equally in both the x and z direction. However, in the D=15 case, the z tip mode is dominating, so I should have less trouble there as well. But that move is shorter in duration as well. The problem is trying to separate all of the effects of the different interacting nonlinearities.

Regular settling times, tsc (sec) Payload=7500

thelbf	accel	tce	unsh	ZV	ZVm	ZVb	ZV3	neg ZV	neg ZV3	ZVD	ZVD3	neg ZVD	neg ZVD3	neg EI
90	0.0010	185.38	8.27											
	0.0020	180.93	14.22	8.88	13.45	13.12	13.38	12.35	12.80	6.43	11.18	4.32	5.20	5.38
	0.0030	179.45	15.00				14.37		13.35	6.97	7.60	5.37	6.03	
	0.0040	178.70	15.30	9.95	14.70	14.53	14.70	13.37	13.97	7.30	7.92	5.47	12.42	6.42
	0.0050	178.25	15.62				14.85		13.95	7.50	8.15	7.02	12.85	
	0.0060	177.95	15.82				15.00		14.12	7.65	8.35	7.10	12.85	
	0.0080	177.60	16.02	10.67	15.17	14.97	15.17	13.92	14.32	7.80	8.55	8.47	13.05	7.60
	0.0160	177.03	16.35											
tis			0.00	5.66	5.13	5.17	5.12	2.79	2.99	11.32	10.25	8.05	7.00	8.13

Envelope settling times, tse (sec) Payload=7500

thelbf	accel	tce	unsh	ZV	ZVm	ZVb	ZV3	neg ZV	neg ZV3	ZVD	ZVD3	neg ZVD	neg ZVD3	neg EI
90	0.0010	185.38	2.29											
	0.0020	180.93	12.84	5.66	7.09	6.32	6.90	9.74	9.77	11.32	10.25	8.05	7.00	8.13
	0.0030	179.45	13.58				12.81		12.25	11.32	10.25	8.05	7.00	
	0.0040	178.70	14.27	7.40	13.53	13.77	13.46	12.86	12.70	11.32	10.25	8.05	7.00	8.13
	0.0050	178.25	14.46				13.91		13.84	11.32	10.25	8.05	7.00	
	0.0060	177.95	14.59				13.93		14.15	11.32	10.25	8.05	8.88	
	0.0080	177.60	14.81	8.35	13.90	14.20	13.91	14.11	14.49	11.32	10.25	8.05	7.53	8.13
	0.0160	177.03	15.10											
tis			0.00	5.66	5.13	5.17	5.12	2.79	2.99	11.32	10.25	8.05	7.00	8.13

Table 5.11: Settling times for P=7500, D=90

5.5.4 *Nominal Payload Results*

For the nominal payload DRS model, the envelope settling times are all equal to the shaper duration, which indicates that input shaping is getting rid of all the residual vibration. Therefore, the fastest shaper is the best. These results also mean that we must actually look at the regular settling times to see when the responses actually get within the settling band.

Since the fastest shaper is giving the biggest increase in performance, some type of ZV shaper should be used. The unshaped case settles within 6 seconds after the end of the command, which is about 3 cycles of vibration because the frequencies are around 0.5 Hz. The fastest and most sensitive shaper does the best for this payload. The frequencies are higher, so small errors in frequencies do not have as large an effect here as they do for the heavier payloads where the frequencies are an order of magnitude lower.

For the P=0, D=15 case, results are shown in Table 5.12. The negative ZV shaper does the best overall. There does not seem to be much vibration here, or at least there is a large amount of damping present, so it dies out fairly quickly, within 3 cycles, so the fastest shaper is doing the best job. The ZVm does worse than the ZV, which shapes for end frequency. Here the z tip position is driving the response, while the x tip is vibrating, but not as much. The z and x frequencies and dampings are the same.

Regular settling times, tsc (sec) Payload=0

thelbf	accel	tce	unsh	ZV	ZVm	neg ZV	ZVD	neg ZVD	neg EI
15	0.0160	8.18	3.50	0.12	0.88	0.62	1.07	0.60	1.00
	0.0320	6.42	6.08	2.08	0.42	0.25	2.30	2.05	2.83
	0.0400	6.08	7.40	2.50	1.22	1.12	2.37	1.80	1.05
	0.0480	5.85	7.55	2.75	1.58	1.73	2.40	1.90	2.93
	0.0640	5.55	6.73						
	tis		0.00	1.32	1.10	0.65	2.64	1.87	1.89

Envelope settling times, tse (sec) Payload=0

thelbf	accel	tce	unsh	ZV	ZVm	neg ZV	ZVD	neg ZVD	neg EI
15	0.0160	8.18	3.23	1.32	1.10	0.65	2.64	1.87	1.89
	0.0320	6.42	6.21	1.32	1.10	0.65	2.64	1.87	1.89
	0.0400	6.08	7.20	1.32	1.10	0.65	2.64	1.87	1.89
	0.0480	5.85	7.34	2.63	1.10	0.65	2.64	1.87	1.89
	0.0640	5.55	7.33						
	tis		0.00	1.32	1.10	0.65	2.64	1.87	1.89

Table 5.12: Settling times for P=0, D=15

Table 5.13 shows the results for the P=0 and D=45 case where the ZV does the best for lower accelerations and the negative ZV does the best for higher accelerations. Shaping for the middle frequency does not do better than shaping for the end frequency. The x tip and z tip are vibrating equal amounts at the same frequencies and dampings.

Regular settling times, tsc (sec) Payload=0

thelbf	accel	tce	unsh	ZV	ZVm	neg ZV	ZVD	ZVDm	neg ZVD	neg ZVDm	neg EI	neg EIm
45	0.0040	28.02	-0.95									
	0.0080	21.02	1.98	-0.02	0.83	0.53	0.78	0.60	0.33	0.35	0.50	0.33
	0.0160	17.52	4.58	0.23	0.78	0.60	1.33	0.93	0.88	0.62	1.33	0.73
	0.0320	15.78	4.42	2.30	0.45	0.22	2.20	2.42	1.22	1.90	2.60	2.22
	0.0400	15.43	6.97	2.75	0.45	0.20	2.20	2.68	3.20	2.02	2.47	3.77
	0.0480	15.20	7.17									
	0.0640	14.90	7.40									
tis			0	1.35	1.06	0.6	2.7	2.12	1.91	1.44	1.93	1.45

Envelope settling times, tse (sec) Payload=0

thelbf	accel	tce	unsh	ZV	ZVm	neg ZV	ZVD	ZVDm	neg ZVD	neg ZVDm	neg EI	neg EIm
45	0.0040	28.02	0.00									
	0.0080	21.02	0.99	1.35	1.06	0.60	2.70	2.12	1.91	1.44	1.93	1.45
	0.0160	17.52	4.08	1.35	1.06	0.60	2.70	2.12	1.91	1.44	1.93	1.45
	0.0320	15.78	4.13	1.35	1.06	0.60	2.70	2.12	1.91	1.44	1.93	1.45
	0.0400	15.43	7.32	1.35	1.06	0.60	2.70	2.12	1.91	1.44	1.41	2.72
	0.0480	15.20	7.43									
	0.0640	14.90	7.76									
tis			0	1.35	1.06	0.6	2.7	2.12	1.91	1.44	1.93	1.45

Table 5.13: Settling times for P=0, D=45

For the P=0 and D=90 case, the ZV shaper does the best job. There is not much vibration in the z tip response, only in the x direction, so the ZV shaper is only really working on the x tip. It does not have to deal with vibrations in both directions, in or out of phase. Table 5.14 gives these results.

Regular settling times, tsc (sec) Payload=0

thelbf	accel	tce	unsh	ZV	ZVm	neg ZV	ZVD	neg ZVD	neg EI
90	0.0020	56.05	-1.18						
	0.0040	42.03	-0.78						
	0.0080	35.03	0.37						
	0.0160	31.53	1.47	-0.05	0.07	-0.23	0.55	0.37	0.37
	0.0320	29.78	-0.40	0.32	0.45	0.12	1.70	0.57	0.57
	0.0480	29.20	4.42	0.25	0.52	1.72	1.07	0.97	0.97
	0.0640	28.90	4.60	0.75	0.55	2.15	1.15	0.97	0.97
tis			0	0.85	0.95	0.5	1.69	1.16	1.17

Envelope settling times, tse (sec) Payload=0

thelbf	accel	tce	unsh	ZV	ZVm	neg ZV	ZVD	neg ZVD	neg EI
90	0.0020	56.05	0.00						
	0.0040	42.03	0.00						
	0.0080	35.03	0.00						
	0.0160	31.53	1.08	0.85	0.95	0.50	1.69	1.16	1.18
	0.0320	29.78	0.00	0.85	0.95	0.50	1.69	1.16	1.17
	0.0480	29.20	4.26	0.85	0.95	0.50	1.69	1.16	1.17
	0.0640	28.90	5.05	0.85	0.95	1.59	1.69	1.16	1.17
tis			0	0.85	0.95	0.5	1.69	1.16	1.17

Table 5.14: Settling times for P=0, D=90

The ZV shaper family does the best job for this payload. A ZV or negative ZV only takes 1.2 or 0.6 seconds, yet in general the shaped responses settle on the way into the settling band. You also get a better % time savings because the moves are so much shorter in duration. For the short D=15 moves, you can get a 45% time savings by using input shaping; for the longer D=90 moves, you get less time savings, only around 12%. Input shaping saves a few seconds. The big benefit is being able to move at fast accelerations to get there quickly and have little vibration. For example, D=45, a=0.004, the arm gets there and settles in 27 seconds without input shaping. With a=0.040, an order of magnitude higher, and a negative ZV shaper, the arm gets there and settles within 15.6 seconds. This is only a 12 seconds difference, but added up over many such moves the overall time savings could make a difference.

5.5.5 Large Payload Results

The 32000 lb. payload case is a puzzling one. Table 5.15 shows the results for a variety of accelerations. Input shaping does not appear to be helping here at all. The ZV shaper is better in a few cases than the unshaped, but not by much. The ZVD, negative ZVD, and negative EI shapers all do very poorly. One potential cause is saturation of the command, but the amplitude of vibration does appear to be increasing with acceleration. Another potential problem is the accuracy of the shaper frequency. If the frequency is off by a hundredth of a Hertz, that is a 25% change in frequency. The negative EI shaper has a very wide sensitivity curve, yet did worse than the ZV. So it seems that the time delay associated with the shaper is having more of an effect than the error in shaper frequency. A multiple mode shaper was tried; however, a ZV/ ZV shaper for two modes would last 23.3 seconds. The time delay is as long as it takes the unshaped cases to settle. shaping for both beginning and end frequencies was also tried, but the ZV_b did worse than the regular ZV. The shapers are simply delaying the move, not allowing it to settle more quickly.

Regular settling times, tsc (sec) Payload=32000

thelbf	accel	tce	unsh	ZV	ZV3	ZVb	neg ZV	neg ZV3	ZVD	ZVD3	neg ZVD	neg ZVD3	neg EI
45	0.0002	168.20	14.32	23.65	22.78								
	0.0004	154.18	17.77	26.27									
	0.0005	151.38	30.85	24.62	26.80	26.73	25.02	24.25	31.82	36.05	27.07	31.65	33.82
	0.0006	149.53	30.60	27.67									
	0.0008	147.18	32.52	28.70									
	0.001	145.78	21.70	27.42	28.90	27.67	27.67	26.80	34.10	37.95	29.25	32.65	31.42
	0.002	142.98	23.30	28.42	30.22	28.65	28.65	28.22	35.42	39.05	30.55	34.22	32.57
	0.004	141.58	34.92	29.05	30.72	29.72	29.72	28.97	36.15	39.57	31.28	35.25	33.67
	0.008	140.88	35.38	29.32									
tis			0.00	12.04	12.66	12.75	8.29	7.58	24.08	25.32	17.39	17.64	17.62

Envelope settling times, tse (sec) Payload=32000

thelbf	accel	tce	unsh	ZV	ZV3	ZVb	neg ZV	neg ZV3	ZVD	ZVD3	neg ZVD	neg ZVD3	neg EI
45	0.0002	168.20	21.54	26.65	27.39								
	0.0004	154.18	24.54	36.50									
	0.0005	151.38	22.33	32.97	32.14	35.87	34.92	34.92	36.65	43.38	28.95	35.04	37.59
	0.0006	149.53	24.03	38.65									
	0.0008	147.18	26.36	39.67									
	0.001	145.78	27.05	28.40	40.12	39.04	33.45	33.45	35.69	41.63	34.76	40.88	31.47
	0.002	142.98	28.51	29.76	42.00	41.69	40.31	40.31	36.79	41.77	37.21	38.40	33.11
	0.004	141.58	29.13	32.06	36.25	42.18	39.30	39.30	42.95	41.64	35.84	42.94	39.58
	0.008	140.88	33.84	32.55									
tis			0.00	12.04	12.66	12.75	8.29	7.58	24.08	25.32	17.39	17.64	17.62

Table 5.15: Settling times for P=32000, D=45

Table 5.16 shows the results for the D=15 and D=90 cases. Input shaping is just delaying the command and the settling time again.

Regular tsc (sec) P=32000

	D=15			D=90		
amax	tce	unsh	neg EI	tce	unsh	neg EI
0.0005				291.57	23.03	26.72
0.0010	52.33	24.08	37.72	285.98	25.92	25.77
0.0020	49.52	25.77	39.30	283.18	27.20	28.03
0.0040	48.12	26.62	40.58	281.77	27.83	28.60
0.0080	47.43	26.98		281.07	28.20	
tis		0.00	19.58		0.00	15.32

Envelope tse (sec) P=32000

	D=15			D=90		
amax	tce	unsh	neg EI	tce	unsh	neg EI
0.0005				291.57	17.86	26.87
0.0010	52.33	33.89	39.28	285.98	20.82	26.36
0.0020	49.52	35.26	41.13	283.18	23.04	24.66
0.0040	48.12	36.01	38.56	281.77	23.81	26.20
0.0080	47.43	36.43		281.07	24.21	
tis		0.00	19.58		0.00	15.32

Table 5.16: Settling times for P=32000, D=15 and D=90

In order to investigate the reasons why input shaping was not working, I took motor friction and stiction out of the DRS model. The results are shown in Table 5.17. Taking friction out did decrease the settling times, but did not improve the input shaping performance. So there seems to be some other nonlinearity hurting input shaping's performance.

Regular tsc (sec) Friction=0 Payload=32000

thelbf	amax	tce	unsh	ZV
45	0.0005	151.38	17.62	26.00
	0.001	145.78	21.20	28.08
	0.002	142.98	22.82	28.52
	0.004	141.58	23.62	29.67
	tis		0	13.84

Envelope tse (sec) Friction=0 Payload=32000

thelbf	amax	tce	unsh	ZV
45	0.0005	151.38	18.75	30.60
	0.001	145.78	22.68	33.31
	0.002	142.98	24.02	36.79
	0.004	141.58	24.93	36.83
	tis		0	13.84

Table 5.17: Settling time for P=32000, D=45, and no friction

5.5.6 *Damping Investigation*

Damping is affected by many different things and it changes a lot in real systems. Precise machines whose performance depends on an accurate system model may recalibrate damping once or twice per hour depending on the application. Damping also tends to be quite temperature sensitive. In space this is a large issue since temperature extremes are common.

To investigate this issue, a series of tests were run using the DRS model with various levels of damping. The damping of the system was changed by changing the motor friction and stiction levels. *n* is defined to be the normal amount of friction and stiction in the DRS, which is a motor friction torque of 0.0208 foot-pounds on each joint, and a motor stiction torque of 0.0208 foot-pounds on each joint. Runs were done for the range of 0, 0.1*n*, 0.5*n*, 1*n*, 2*n*, and 5*n*. Table 5.18 gives the unshaped results.

Regular tsc (sec), P=7500, a=0.004

friction	d=15	d=45	d=90
0n	14.12	13.00	9.90
0.1n	14.15	12.95	9.80
0.5n	13.85	12.75	14.10
n	20.90	19.55	15.30
2n	21.37	19.85	20.67
5n	35.33	33.33	31.80
tce	31.63	90.45	178.70

Envelope tse (sec), P=7500, a=0.004

friction	d=15	d=45	d=90
0n	15.62	14.62	11.54
0.1n	15.82	14.23	11.84
0.5n	17.14	15.64	12.46
n	19.70	18.37	14.27
2n	23.42	22.27	17.67
5n	35.93	34.19	29.67
tce	178.70	90.45	31.63

Table 5.18: Settling times for varying levels of friction, P=7500, a=0.004

The results were all about the same for friction levels of 0, 0.1*n*, and 0.5*n*. The settling time function just calculated the time to settle to the final position, not the final desired position, so it ignores the fact that the moves go to different end positions. There is a large jump at 5*n*, with settling times twice as long as at the lower friction levels. So for the lower range, the friction does not dominate the results, but once you get above a certain level, the friction dominates. The higher friction makes the vibrations lasts longer. This result directly contradicts the study done in Chapter 2 on friction. There we saw that additional friction added damping to the system and the amplitudes of the residual vibrations were reduced. In the SRMS system, which is highly nonlinear, the opposite effect is being seen.

The time histories shown in Figure 5.17 shows the P=7500, a=0.004 rad/s^2, D=45 degrees case for a variety of friction levels. The friction levels shown here are f=0, f=0.5*n*, f=2*n*, and f=5*n*.

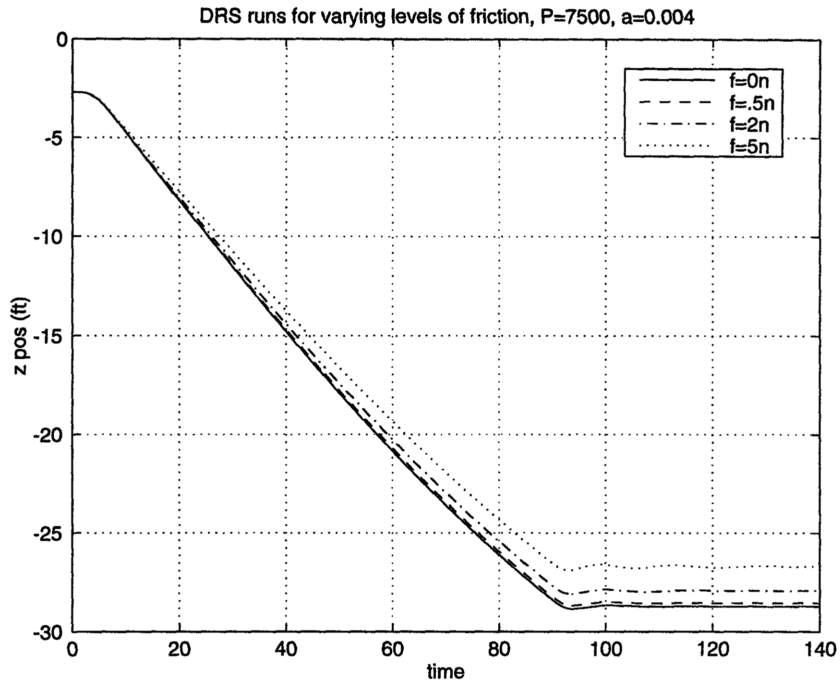


Figure 5.17: DRS varying friction runs, tip z position

The friction is changing the system response by making it move slower. It is also causing more vibration at the end of the move, as can be seen in Figure 5.18. More stiction and friction seems to lead to more residual vibration in the joints.

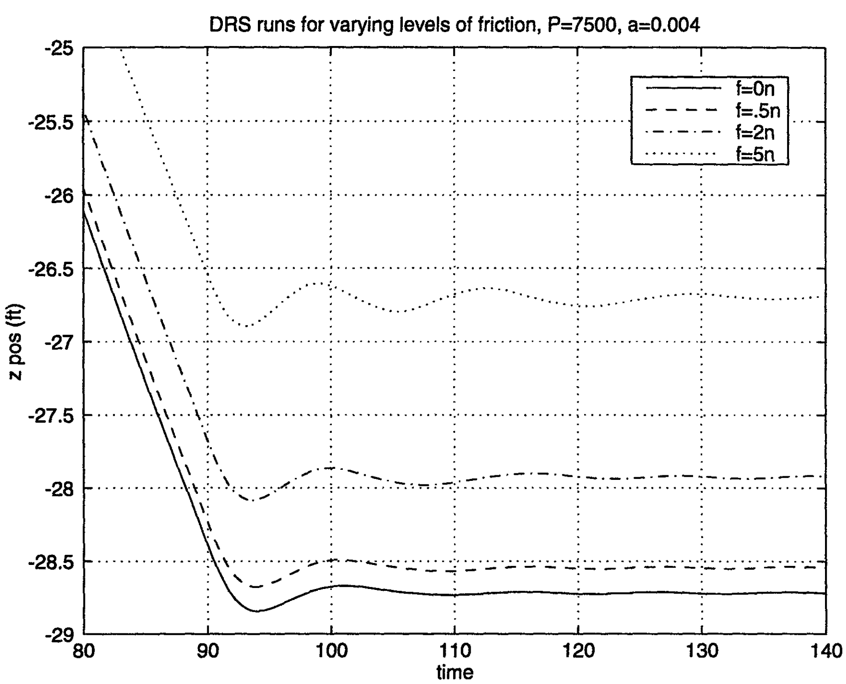


Figure 5.18: Detail of DRS varying friction runs, P=7500

One possible explanation for the increase in residual vibration is that there is a different method of energy dissipation. For normal levels of friction and stiction, energy is dissipated in the joints and in the links. When the very high levels of friction are added to the system, the joints might lock and not be able to dissipate any energy there. Then all of the excess energy would have to be dissipated in the links. This would cause the links to vibrate longer and with higher amplitude. The link flexibility is defined to be the tip position minus the rigid body position, which is the position that the tip would be at if the links were perfectly rigid. When the link flexibility was plotted for several friction cases, the links are definitely oscillating more at the higher friction levels. So this explanation is plausible. However, more evidence is needed to prove this theory.

Regular tsc (sec), P=7.5K, a=0.004, D=45

friction	tce	unsh	ZV	ZVD	ZVD3	neg ZVD2
0.00	90.45	13.00	16.35	15.77	18.35	14.12
0.1n	90.45	12.95	16.67	15.80	18.45	14.17
0.5n	90.45	12.75	17.40	15.77	18.60	14.12
n	90.45	19.55	17.62	15.77	18.75	13.92
2n	90.45	19.85	24.70	15.52	18.67	21.20
5n	90.45	33.33	31.52	14.10	9.85	14.33
tis		0.00	5.54	11.09	13.33	7.53

Envelope tse (sec), P=7.5K, a=0.004, D=45

friction	tce	unsh	ZV	ZVD	ZVD3	neg ZVD2
0.00	90.45	14.62	15.28	15.19	16.26	14.43
0.1n	90.45	14.23	15.44	15.25	16.52	14.65
0.5n	90.45	15.64	16.49	16.06	17.16	15.40
n	90.45	18.37	18.66	16.76	20.36	16.37
2n	90.45	22.27	22.20	17.73	16.78	18.29
5n	90.45	34.19	44.36	11.09	13.33	14.74
tis		0.00	5.54	11.09	13.33	7.53

Table 5.19: Interactions between friction and input shaping, P=7500, D=45, a=0.004

Input shaping was implemented on the 7500 lb. payload system with various levels of friction, as shown in Table 5.19. The results from the ZVD shaper are very consistent; all settling times are around 15.7 seconds, no matter what level of friction. In terms of improvement in performance, we do see improvement at the higher friction levels, but at the lower levels, the unshaped case is doing better. The negative ZVD shaper has the most best performance for the low friction levels. For the 5*n* case, the shaped response does not vibrate as much as the unshaped response. In fact, the ZVD response looks like the lower friction responses. Since the links are vibrating to dissipate more energy and input shaping reduces the amount of energy put into the system modes, there should be less excess energy to dissipate when input shaping is used.

The less friction there is, the quicker the move reaches its desired position and settles, as can be seen in the previous plots. Input shaping does better for the higher friction levels, but that is partially because I was using the frequency from the normal levels of friction runs. As the friction level increases, the residual vibration's frequency and damping appears to decrease. When the friction equals 0.1*n*, the residual vibration frequency is 0.082 Hz and the damping is 0.237. At friction equal to 5n, the frequency is 0.069 Hz and the damping is 0.128.

The energy dissipation mechanism appears to change the system characteristics a little.

The ZVD family of input shapers is doing better than the ZV shaper here, but the same frequency was used for all input shapers so any results are only valid for the close-to-normal friction levels.

5.5.7 Conclusions

I am not seeing completely consistent patterns within each move distance and payload. For example, if one input shaper did the best for a certain move distance, payload, and acceleration, I would expect it to do the best job for all accelerations. However, this is not the case. A family of input shapers dominates; for example, in the P=7500 and D=45 degree case, the ZVD and negative ZVD shapers do the best. The DRS results are summarized in Table 5.20.

Payload	0	7500	32000
D=15	ZV, neg ZV	ZVD, neg ZVD	unshaped
D=45	ZV, neg ZV	ZVD, neg ZVD	unshaped
D=90	ZV	neg ZVD, ZVD	unshaped

Table 5.20: Summary table for DRS data

The unloaded cases are the easiest to analyze. A simple ZV or negative ZV input shaper will shave several seconds off the settling time. The realization here is that you can move the unloaded arm very quickly, if you trust your software. At an acceleration of 0.040 rad/s^2 and a maximum velocity of 3.21 deg./s, the SRMS can complete a 90 degree elbow joint move in 30 seconds if a ZV shaper is used.

With a midsized payload the SRMS must move much more slowly. The saturation of the motor means that it is limited to an acceleration of 0.005 rad/s^2. It takes 190 seconds to complete the same 90 degree elbow move, even if input shaping is added to reduce the vibration. A motor with a larger allowable torque is needed to speed the joints up. The more robust input shapers such as ZVD and negative ZVD give the most performance increase for this case. However, the moves are long enough that for D=45, the best time savings you get is 13%. The unshaped moves take around 2 cycles to settle, so that does not give input shaping much room to improve the system performance.

The heavy 32000 lb. payload was resistant to input shaping. When joint friction and stiction were taken out of the DRS model, the unshaped responses settled more quickly, but input shaping did not help any more than usual. The nonlinearities of the DRS are more apparent with this extremely large payload and inertia.

One of the things these studies are showing is that the workspace of the RMS has to be very well mapped for frequencies. Errors in system knowledge are hindering the input shapers' performance.

5.6 Conclusions

A simple two-link model was developed and compared with the $FLEX_2$ SRMS rate controller simulation. It was tested using a trapezoidal trajectory profile to see how residual vibration changes with acceleration, move distance, and end position. The DRS simulation was used to complete the study of changing geometry problem. Three different payloads were examined to see the differences in the workspace. Input shaping was added to the system to see if it could reduce the residual vibrations. A different input shaper gave the best performance for each payload. A ZV or negative ZV shaper did the best job for the unloaded cases, where frequencies were the highest and did not shift as much. The ZVD family of input shapers gave the most performance improvement for the midsize payload cases. The heaviest payload did not respond well to input shaping; a ZV shaper only helped in a few cases. The ZV shaper has the least amount of time delay and does the best of the shapers tried; however it does not do very well. The associated time delay hurts the shapers a large amount here because the frequencies are so low.

The heavy payload case is not getting a performance increase by using input shaping. However, the unshaped response has a long settling time so there is room for improvement. An advanced feedback controller like those designed in Chapter 4 should offer a large performance increase. The controller would allow the system to settle as soon as the command finishes or before. The controller would have to be designed for a range of frequencies since the geometry changes, but that can be done and was done for the FLEX controllers.

In general, using a trapezoidal trajectory is recommended. It has a much smoother profile than just a step command and excites much less vibration. If you calculate the length of the acceleration, you can get the phase of the deceleration 180 degrees out of phase with the acceleration vibrations and they will cancel. However, it is not simple to get the trajectory to do this and when it does happen, it usually restricts you to certain low velocities and accelerations.

The nonlinearities associated with the complex SRMS make it difficult to separate out the effects. Friction was investigated to see how changing its levels changed the system damping and responses. When the motor friction levels were increased, the system had more residual vibration. One explanation is that the energy dissipation mechanism might be changing when the friction is increased. If energy is dissipated mostly in the joints when the friction is at normal levels, when friction is increased it might be locking the joints and preventing energy dissipation. Then the links would have to vibrate more to dissipate the excess energy. This theory was supported by plots of the flexibility of the links which showed that the links were vibrating more for the high friction cases.

Part of the original hypothesis was verified. For a sensitive system, which means that the system frequencies change by at least 20% during the move, and moves longer in duration than five cycles of vibration, the unshaped SRMS response settled to within a radius of two inches immediately. No input shaping is necessary if the desired radius is that large, which validates some of the hypothesis. However, neither the DRS or the $FLEX_2$ models settled during the initial rise if a smaller radius of 0.5 inches was used. This indicates that there is residual vibration of small amplitude. For precise positioning tasks, it is not unreasonable to need to settle to within a very small radius. Therefore, some additional method is needed to reduce the residual vibration.

Conclusion

Chapter 6

6.1 Summary

In this thesis I have investigated the effects of system damping, friction, and control strategies on flexible space systems. The main system investigated was the Shuttle Remote Manipulator System (SRMS). A detailed three-dimensional model of it was created during the FLEX program. A simple two-link model was developed that approximated its behavior. The Draper Remote Manipulator Simulation was also used to simulate the SRMS. Each of these models incorporated different nonlinearities, and thus the results were slightly different from each.

In Chapter 2, input shaping was explained and the results of two small studies were presented. The system damping study explored the interaction between damping and input shapers. Once the damping ratio gets above 0.7, input shaping does not save much time because the unshaped response settles faster than the shaped responses. As the modeling error gets larger, a more insensitive shaper is needed for the lower damping ratios. As the settling band increases, more residual vibration is allowed, so the sensitive shapers perform the best. Tables were generated to allow someone to pick the best input shaper to use if they know the approximate damping ratio of the system, the allowable amount of residual vibration, and the uncertainty of the system knowledge.

The friction study showed that a simple nonlinearity like friction does not impair the input shaper's performance much at all. Even with varying friction magnitudes it was possible for a single mode shaper to obtain a 96% vibration reduction compared to the closed loop, proportional controller step response. The multiple mode shapers performed even better and reduced the vibration to well below the limiting threshold of 0.001 except near instability points. In this study, the additional friction reduced the vibration amplitude and made the responses settle faster.

The background and motivation of the FLEX program was explained in Chapter 3. The FLEX program was designed to develop a better control strategy for the space arms that will be used for construction. A model of the SRMS was developed during Phase A. Many feedback and feedforward methods were initially investigated, but only three feedback controllers were chosen to be pursued in Phase B.

Chapter 4 discusses the results generated during the FLEX program. The best feedback controller turned out to be the gain-scheduled LQG, GSLQG. However, when input shaping was added to the GSLQG, it had even better performance in the small moves. When input shaping was added to the SRMS rate controller, its settling time decreased to a third of the unshaped value. It still took twice as long to settle as the GSLQG plus ZV input shaper, but it saved a lot of controller effort and calculation time.

The focus of Chapter 5 was exploring the problem of changing geometry systems. The SRMS's frequencies change as it moves through the workspace. Many tests were done along different axes of the test matrix to find the best solution to this problem. A simple two-link model was created to represent the SRMS. The model was benchmarked against the $FLEX_2$ model and the first system frequencies matched very well. Both responses settled in the same amount of time. The two-link and $FLEX_2$ models were tested using trapezoidal velocity profiles. Their behaviors were similar, but the saturation and friction in the $FLEX_2$ model restricted its maximum velocities and acceleration. When input shaping was added to the two-link model, the system performance did improve.

The DRS model was used to do many simulations for a variety of distances, accelerations, and input shapers. For the unloaded arm, a ZV input shaper gave the most performance increase and had a time savings of 12 to 45%. For the 7500 lb. payload, a ZVD or negative ZVD was the best input shaper to use and had a time savings of 11 to 24%. The heavy 32000 lb. payload did not respond well to input shaping. A ZV shaper only helped in certain cases. When friction was taken out of the DRS model, input shaping still did not help the 32000 lb. case.

For the heavy payload case, the settling time after the command finishes is around 30 seconds, which is two cycles of vibration. It seems like a good feedback controller would help the system performance for this payload. The current rate controller does not go to the desired final position or settle quickly. Adding a few strain gauges to the links would allow a more advanced controller to be used. As was shown in Chapter 4, a well-designed feedback controller can make the system respond and settle very quickly for any type of input. Since input shaping is not improving performance for the heavy payload case, implementing a better feedback controller designed for heavy payloads should be investigated.

Adding more friction to the 7500 lb. payload case caused the arm to move slower and have more residual vibration. This contradicts the results seen in Chapter 2. The additional friction and stiction do slow down the system response. For the same commands, the higher friction responses do not move as far or follow the command as well as the lower friction cases. One possible explanation for this behavior is that the high friction and stiction levels are locking the joints so that the joints can no longer dissipate energy. Instead the flexible links are the only mechanism for dissipating the energy. Thus the links vibrate more and longer to get rid of the excess energy. Input shaping was effective at reducing vibration for the higher friction levels. The shaped responses look like the lower friction unshaped responses. Input shaping does not input as much energy into the system, so there is less energy to dissipate at the end and the links do not have to oscillate as much to get rid of the energy.

The DRS is a very useful model of the SRMS. Interactions between the rate controller and input shaping were investigated. In most cases input shaping was helpful and improved performance. However, there was no one best input shaper. Different input shapers work best with different payloads and in different areas of the workspace.

6.2 Future Work

The FLEX program is planned to be a four phase program. Consequently, we have only begun to look at the SRMS and different control techniques. Future phases of the program will involve designing a flexible manipulator to work in the Shuttle middeck. New controllers must be developed to work with this new, smaller manipulator. The controllers also must be designed to work in a three-dimensional workspace. The three-dimensional FLEX SRMS model was developed near the end of Phase A. Therefore, no controllers were designed to take advantage of the 3D system. The planar controllers were just imported into the 3D FLEX model and they worked as well as before. However, there was only a rate controller on the shoulder yaw joint, so the overall performance was degraded. The control problem needs to be addressed. When an astronaut is added to the control problem, the results grow even more complicated. The interactions between an automatic feedback controller and an astronaut need to be investigated further, preferably on orbit. Hopefully this will be accomplished during future phases of the FLEX program.

Further work needs to be done on both the three-dimensional control problem and the input shaping problem. When you are shaping for three-dimensional moves, the modes are usually not the same in all directions. Certain modes have zero amplitudes in certain directions or vibrate between out-of-plane. Since the frequencies are changing over the course of the move, the problem grows even more complex. Input shaping was derived as a method for dealing with one-dimensional motion. There is no methodology of how to apply input shaping in two or three dimensions. Multiple mode shapers can be used, but it is unclear whether to shape for joint vibrations or tip vibrations. Also, where should you think about input shaping. Ideally, you want to eliminate vibrations at the tip of the arm. Yet the inputs to the system are joint commands. The noncolocated sensors and actuators problem needs to be explored more fully.

The DRS simulation proved surprisingly easy to run and modify, once a few changes were made to the code to make it easier to read in initial angles and commands. Now that we have realized how simple it is to run, further work can focus on exploring all the different nonlinearities incorporated into the DRS model. The moves were only done in a single plane, but the DRS is also three-dimensional, so more work need to be done to explore the entire workspace. If different feedback controllers could be added to the DRS, even more discoveries could be made.

Payload mass had a large effect on the results in Chapter 5. The unloaded arm can be controlled well with just the Rate controller and input shaping. The heavier payload case poses the more challenging problems. Future work should focus on these problems where the system frequencies are very low and moves very long. There must be better ways to move these payloads around. Another module of the DRS that I did not explore is the flexible payload module. It has the ability to create and simulate many different flexible payloads, of varying mass and stiffness. If we are planning on making recommendations about better ways to control the SRMS during space construction, a flexible payload could represent some of those tasks better than models of a stiff cylinder.

I proposed a hypothesis in Chapter 5 about the relationship between frequencies and move duration. I think another factor that should be investigated is system damping. The SRMS models I have been using have a high enough damping that unshaped vibrations die out in about two cycles. A more lightly damped system would have very different behavior and system response. The hypothesis should be tested for more systems, ideally for a system which has a large frequency change during a short move so that systems of type III.B can be tested, if they exist outside of simulations. The Flexbot, a two-link flexible robot, should also be used to test the hypothesis experimentally. Its frequencies change as it moves through the workspace, and it was designed to represent the SRMS, so it should be a type III.A system with actual backlash, friction, and other nonlinearities.

References

1. Ardayfio, David D. *Fundamentals of Robotics.* New York: Marcel Dekker, Inc., 1987.

2. Banerjee, Arun K. and Singhose, William E. "Slewing and Vibration Control of a Nonlinearly Elastic Shuttle Antenna," *AIAA Guidance, Navigation, and Control Conference,* Scottsdale, AZ, 1994.

3. Bederson, Benjamin B., Wallace, Richard S., and Schwartz, Eric L. "Control and Design of the Spherical Pointing Motor," *Proceedings of the IEEE International Conference on Robotics and Automation,* Atlanta, 1993, p. 630-636.

4. Carusone, Joseph and D'Eleuterio, Gabriele M. T. "Experiments in the Control of Structurally Flexible Manipulators with the RADIUS Facility," *Proceedings of the Second Joint Japan/US Conference on Adaptive Structures,* Nagoya, Japan, 1991, p. 589-605.

5. Chang, Ken W. *Shaping Inputs to Reduce Vibration in Flexible Space Structures,* Master of Science Thesis, Massachusetts Institute of Technology, 1992.

6. Christian, Andrew D. *Design and Implementation of a Flexible Robot,* Massachusetts Institute of Technology Artificial Intelligence Laboratory Technical Report No. 1153, 1989.

7. Drapeau, Vincent and Wang, David. "Verification of a Closed-loop Shaped-Input Controller for a Five-bar-linkage Manipulator," *Proceedings of the IEEE International Conference on Robotics and Automation,* Atlanta, 1993, p. 216-221.

8. Feddema, J.T. "Digital Filter Control of Remotely Operated Flexible Robotic Structures," *Proceedings of the American Control Conference,* San Francisco, 1993, p 2710-2715.

9. Hillsley, Kenneth L. and Yurkovich, Stephen. "Vibration Control of a Two-Link Flexible Robot Arm," *Proceedings of IEEE International Conference on Robotics and Automation,* Sacramento, CA, 1991, p 2121-2126.

10. Hollars, Michael G. and Cannon, Robert H. "Initial Experiments on the End-point Control of a Two-Link Manipulator with Flexible Tendons," *ASME Winter Annual Meeting*, November 1985.

11. Hyde, James M. *Multiple Mode Vibration Suppression in Controlled Flexible Systems*, Master of Science Thesis, Massachusetts Institute of Technology, 1991.

12. Jones, Steven D. and Ulsoy, A. Galip. "Control Input Shaping for Coordinate Measuring Machines," *Proceedings of the American Controls Conference*, Baltimore, Maryland, 1994, p 2899-2903.

13. Magee, David P. and Book, Wayne J. "Eliminating Multiple Modes of Vibration in a Flexible Manipulator," *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, GA, p 474-478.

14. Magee, David P. and Book, Wayne J. "Filtering Schilling Manipulator Commands to Prevent Flexible Structure Vibration," *Proceedings of the American Control Conference*, Baltimore, MD, 1994, p. 2538-2542.

15. Meckl, Peter H. and Kinceler, Roberto. "Trajectory Determination for Vibration-free Motions of a Flexible-Joint Robot," *Proceedings of the American Control Conference*, Baltimore, MD, 1994, p. 2521-2525.

16. Metzinger, Richard W. *DRS Users Guide.* Charles Stark Draper Laboratories, 1985.

17. Newsome, J., Layman, W., Waites, H., and Hayduk, R. "The NASA Controls-Structures Interaction Technology Program," *Proceedings of the 41st Congress of International Astronautical Foundation*, Dresden, Germany, 1990.

18. Oakley, Celia M. and Cannon, Robert H. "Anatomy of an Experimental Two-Link Flexible Manipulator under End-point Control," *Proceedings of the 29th Conference on Decision and Control*, Honolulu, HI, 1990, p. 507-513.

19. *Payload Deployment and Retrieval System Simulation Database Version 1.0*, JSC-25134, 1991.

20. Rappole, Bert W. *Minimizing Residual Vibrations in Flexible Systems*, Master of Science Thesis, Massachusetts Institute of Technology, 1992.

21. Rogers, Keith E. *Limiting Vibration in Systems with Constant Amplitude Actuators through Command Preshaping*, Master of Science Thesis, Massachusetts Institute of Technology, 1994.

22. Schmitz, Eric and Ramey, Madison. "Initial Experiments on the end-point control of a 2-DOF long-reach elastic manipulator," *Proceedings of the SPIE Symposium on Advances in Intelligent Systems*, Boston, MA, 1991.

23. Seering, Warren. "Control of Flexible Construction Systems: the In-Step Phase A final report", 1995.

24. Singer, N.C. and Seering, W.P. "Preshaping Command Inputs to Reduce System Vibration," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 112, March 1990, p. 76-82.

25. Singer, Neil C. *Residual Vibration Reduction in Computer Controlled Machines*, Doctoral Thesis, Massachusetts Institute of Technology, 1989.

26. Singhose, William, Singer, Neil, and Seering, Warren. "Comparison of Command Shaping Methods for Reducing Residual Vibration," *Third European Controls Conference*, 1995.

27. Singhose, William, Singer, Neil, and Seering, Warren. "Design and Implementation of Time-Optimal Negative Input Shapers," Submitted to the Journal of Dynamic Systems, Measurement, and Control, 1994.

28. Singhose, William E. *Shaping Inputs to Reduce Residual Vibration: A Vector Diagram Approach*, Bachelor's Thesis, Massachusetts Institute of Technology, 1990.

29. Smith, O.J.M. *Feedback Control Systems*, New York: McGraw-Hill Book Company, Inc., 1958.

30. Tzes, Anthony and Yurkovich, Stephen. "An Adaptive Input Shaping Control Scheme for Vibration Suppression in Slewing Flexible Structures," *IEEE Transactions on Control Systems Technology*, Vol. 1, June 1993, p. 114-121.

31. Tzes, Anthony P, Englehard, Matthew J., and Yurkovich, Stephen. "Input Preshaping With Frequency Domain Information For Flexible-Link Manipulator Control," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Boston, MA, 1989, p 1167-1175.

32. Tuttle, T.D. and Seering, W.P. "A Zero-placement Technique for Designing Shaped Inputs to Suppress Multiple-mode Vibration," *Proceedings of the American Controls Conference*, Baltimore, MD, 1994, p. 2533-2537.

33. Yoshikawa, Tsuneo. *Foundations of Robotics: Analysis and Control*. Cambridge, MA: MIT Press, 1990.

34. Zuo, Kai and Wang, David. "Closed Loop Shaped-Input Control of a Class of Manipulators with a Single Flexible Link," *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, 1992, p 782-787.

Extra Data from Input Shaping Studies

Appendix A

A.1 Additional Data from Damping Ratio Study

	2% error		5% error		10 % error		20% error	
damping	best IS	% t saved	best IS	% t saved	best IS	% t saved	best IS	% t saved
0.01	ZVD	98	ZVD	98	ZVDD	97	ZVDDD	96
0.02	ZVD	97	ZVD	97	ZVDD	95	ZVDDD	92
0.03	ZVD	95	ZVD	95	ZVDD	92	ZVDDD	88
0.04	ZVD	94	ZVD	94	ZVDD	90	ZVDDD	85
0.05	ZVD	92	ZVD	92	ZVDD	87	ZVDDD	80
0.06	ZVD	91	ZVD	90	ZVDD	85	ZVDDD	76
0.07	ZVD	89	ZVD	89	ZVD	87	ZVDDD	72
0.08	ZV	89	ZVD	87	ZVD	85	ZVDDD	69
0.09	ZV	87	ZVD	85	ZVD	83	ZVDDD	64
0.10	ZV	86	ZVD	84	ZVD	83	ZVDDD	61
0.15	ZV	82	ZVD	76	ZVD	75	ZVDD	52
0.20	ZV	85	ZVD	69	ZVD	67	ZVDD	44
0.25	ZV	79	ZV	59	ZVD	54	ZVDD	23
0.30	ZV	74	ZV	49	ZVD	42	ZVD	7
0.35	ZV	73	ZV	50	ZV	43	ZVD	26
0.40	ZV	64	ZV	37	ZV	26	ZVD	16
0.45	ZV	63	ZV	63	ZV	26	ZV	16
0.50	ZV	61	ZV	61	ZV	25	ZV	13
0.55	ZV	45	ZV	44	none	0	none	0
0.60	ZV	44	ZV	44	ZV	44	none	0
0.65	ZV	43	ZV	42	ZV	42	none	0
0.70	ZV	40	ZV	40	ZV	40	none	0
0.80	none	0	none	0	none	0	none	0
0.90	none	0	none	0	none	0	none	0

Table A.1: Results for the 2% settling band

Table A.1 shows the best input shaper when a 2% settling band was used. The rows in the table represent the damping ratios and the columns represent different errors in system knowledge. The % time saved is the unshaped settling time minus the shaped settling time all divided by the unshaped settling time. Its highest value is 100, but it can go negative if the shaped settling time is greater than the unshaped. The higher % time saved values are more desirable.

Table A.2 shows the best input shaper for a larger settling band of 5%.

	2% error		5% error		10 % error		20% error	
damping	best IS	% t saved	best IS	% t saved	best IS	% t saved	best IS	% t saved
0.01	ZV	99	ZVD	98	ZVD	98	ZVDDD	95
0.02	ZV	98	ZVD	96	ZVD	96	ZVDD	93
0.03	ZV	97	ZVD	94	ZVD	94	ZVDD	89
0.04	ZV	96	ZVD	92	ZVD	91	ZVDD	85
0.05	ZV	95	ZVD	90	ZVD	90	ZVDD	82
0.06	ZV	94	ZVD	88	ZVD	87	ZVDD	78
0.07	ZV	93	ZVD	86	ZVD	85	ZVDD	74
0.08	ZV	92	ZV	85	ZVD	83	ZVDD	70
0.09	ZV	92	ZV	83	ZVD	81	ZVDD	67
0.10	ZV	91	ZV	82	ZVD	79	ZVDD	63
0.15	ZV	86	ZV	75	ZVD	69	ZVDD	46
0.20	ZV	80	ZV	80	ZV	57	ZVD	52
0.25	ZV	75	ZV	75	ZV	47	ZVD	39
0.30	ZV	73	ZV	73	ZV	45	ZVD	35
0.35	ZV	65	ZV	65	ZV	32	ZV	18
0.40	ZV	63	ZV	63	ZV	63	ZV	16
0.45	ZV	46	ZV	46	ZV	46	none	0
0.50	ZV	46	ZV	46	ZV	46	none	0
0.55	ZV	45	ZV	44	ZV	44	none	0
0.60	ZV	42	ZV	42	ZV	42	ZV	42
0.65	ZV	38	ZV	38	ZV	38	ZV	38
0.70	none	0	none	0	none	0	none	0
0.80	none	0	none	0	none	0	none	0
0.90	none	0	none	0	none	0	none	0

Table A.2: Results for the 5% settling band

A.2 Effects of Friction

Following are the equations of motion for the two mode system:

$$\begin{aligned} m_1\ddot{x}_1 + k_s(x_1 - x_2) &= F_{total} \\ m_2\ddot{x}_2 + k_s(x_2 - x_1) &= 0 \end{aligned} \tag{A.1}$$

Table A.3 gives the constants for all three cases.

	case 1	case 2	case 3
mass 1 (kg)	1	1	1
mass 2 (kg)	2	1	1
spring (N/m)	905	905	905

Table A.3: Mass and spring parameters for three cases

For the linear non-collocated case and collocated case, the total force was calculated by Equations A.2 and A.3 respectively:

$$F_{total} = K(x_{2_{ref}} - x_2) \tag{A.2}$$

$$F_{total} = K(x_{1_{ref}} - x_1) \quad \text{(A.3)}$$

For the nonlinear case, the friction term was added to the total force. Following are the equations for the total force for the non-collocated and collocated cases.

$$F_{total} = K(x_{2_{ref}} - x_2) - F_{friction} \quad \text{(A.4)}$$

$$F_{total} = K(x_{1_{ref}} - x_1) - F_{friction} \quad \text{(A.5)}$$

The friction force is shown graphically in Figure 2.13 as a plot of force vs. velocity of mass 1. It is a smoothed form of Coulomb's friction made by joining a straight line segment of positive slope that passes through the origin to two curves from an ellipse to two horizontal lines. I tried to fit a cubic to this curve, as well as arcs from a circle instead of the ellipse, but the math did not work out. The general line and ellipse equations are given in Equations A.6 a, b, c:

$$y = mx \qquad y = fmag \quad \text{(A.6 a, b, c)}$$

$$\left(\frac{x-h}{a}\right)^2 + \left(\frac{y-k}{b}\right)^2 = 1$$

The point (x_1,y_1) is where $y=mx$ and the ellipse intersect; the point (x_2,y_2) is where the ellipse and $y=f_{mag}$ intersect. If we specify (x_1,y_1), this defines the line $y=mx$ where $m=y_1/x_1$. y_2 must be equal to f_{mag} in order to fit Equation A.6c. We also specify x_2, choosing 2^*x_1. Then we can solve for constants a, b, h, and k by plugging the two points into Equation A.6b. After some algebra we have:

$$h = x_2 \qquad b = y_2 - k \quad \text{(A.7 a, b, c, d)}$$

$$k = \frac{y_2^2 - y_1^2 + my_1(x_1 - h)}{2y_2 - 2y_1 + m(x_1 - h)} \qquad a = \frac{b(h - x_1)}{\sqrt{b^2 - (y_1 - k)^2}}$$

x_1 and x_2 were chosen to be small relative to the velocity: x_1=0.0005 and x_2=0.0010. The slope of the line was chosen to be $m=0.7^*y_2/x_1$, which is rather steep since $y_2=f_{mag}$=2 or 5 or 10 N. In the actual MATLAB code, y_2 is always equal to fmag, but f_{mag} changes between simulations, so the constants h, k, b, and a must be calculated each time the program is run.

Two-link Model Equations

Appendix B

I derived the equations of motion for a two link system using Lagrangian methods. I found it easiest to split the system into two smaller systems and then combine the results at the end. The system consists of two rigid links with an elbow joint connecting them and a payload at the end of the links. Each link has a mass and inertia; the elbow joint and payload have masses and inertias as well. The sub-systems were a two-link system with only link masses and inertias and a two-link system with joint mass and a payload. Full derivations can be found in references [1] and [33].

The first set of equations derives the equations of motion for a two-link system with a mass, M_e, at the elbow joint between link 1 and link 2 and a payload, M_p, at the end of link 2. The associated inertias area also included, I_e and I_p. (x_1, y_1) is the position of the elbow joint and (x_2, y_2) is the position of the payload. θ_1 and θ_2 are the shoulder pitch and elbow pitch angles, respectively. l_1 and l_2 are the lengths of the links.

First the coordinates and velocities of the payload mass are found.

$$x_2 = l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) \quad \dot{x}_2 = -l_1\dot{\theta}_1\sin(\theta_1) - l_2(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) \tag{B.1}$$

$$y_2 = l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) \quad \dot{y}_2 = l_1\dot{\theta}_1\cos(\theta_1) + l_2(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) \tag{B.2}$$

Then the velocity of the payload is calculated.

$$v_2^2 = l_1^2\dot{\theta}_1^2 + l_2^2(\dot{\theta}_1+\dot{\theta}_2)^2 + 2l_1l_2\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_2) \tag{B.3}$$

The Lagrangian is given in Equation B.4.

$$\begin{aligned} L &= \tfrac{1}{2}m_e l_1^2\dot{\theta}_1^2 + \tfrac{1}{2}I_e\dot{\theta}_1^2 + \tfrac{1}{2}m_p l_1^2\dot{\theta}_1^2 + \tfrac{1}{2}m_p l_2^2(\dot{\theta}_1+\dot{\theta}_2)^2 \\ &+ m_p l_1 l_2\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_2) + \tfrac{1}{2}I_p(\dot{\theta}_1+\dot{\theta}_2)^2 \end{aligned} \tag{B.4}$$

Once the derivatives of the Lagrangian are taken, you end up with two equations of motion.

$$\begin{aligned}&\ddot{\theta}_1[(m_e+m_p)l_1^2+I_e+m_pl_2^2+2m_pl_1l_2\cos(\theta_2)+I_p]+\ddot{\theta}_2[m_pl_2^2+m_pl_1l_2\cos(\theta_2)+I_p]\\&-2m_pl_1l_2\dot{\theta}_1\dot{\theta}_2\sin(\theta_2)-m_pl_1l_2\dot{\theta}_2^2\sin(\theta_2)=T_1\end{aligned}\quad\text{(B.5)}$$

$$\ddot{\theta}_1[m_pl_2^2+m_pl_1l_2\cos(\theta_2)+I_p]+\ddot{\theta}_2[m_pl_2^2+I_p]+m_pl_1l_2\dot{\theta}_1^2\sin(\theta_2)=T_2\quad\text{(B.6)}$$

These equations can be combined into matrix form. Note that the mass matrix is symmetric.

$$\begin{aligned}&\begin{bmatrix}(m_e+m_p)l_1^2+I_e+m_pl_2^2+2m_pl_1l_2\cos(\theta_2)+I_p & m_pl_2^2+m_pl_1l_2\cos(\theta_2)+I_p\\ m_pl_2^2+m_pl_1l_2\cos(\theta_2)+I_p & m_pl_2^2+I_p\end{bmatrix}\begin{Bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{Bmatrix}=\\&\begin{Bmatrix}T_1\\ T_2\end{Bmatrix}+\begin{bmatrix}0 & m_pl_1l_2\sin(\theta_2)\\ -m_pl_1l_2\sin(\theta_2) & 0\end{bmatrix}\begin{Bmatrix}\dot{\theta}_1^2\\ \dot{\theta}_2^2\end{Bmatrix}+\begin{bmatrix}2m_pl_1l_2\sin(\theta_2)\\ 0\end{bmatrix}\{\dot{\theta}_1\dot{\theta}_2\}\end{aligned}\quad\text{(B.7)}$$

The second set of equations derives the equations of motion for the two-link system with the masses and inertias of the links included. (x_1, y_1) is the position of the center of mass of the first link and (x_2,y_2) is the position of the center of mass of the second link. We assume that the mass density of the links are evenly distributed so that the mass and inertia of the link acts at L/2.

First the coordinates and velocities of the center of mass of link 2 are found.

$$x_2=l_1\cos(\theta_1)+\tfrac{1}{2}l_2\cos(\theta_1+\theta_2)\;\;\dot{x}_2=-l_1\dot{\theta}_1\sin(\theta_1)-\tfrac{1}{2}l_2(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2)\quad\text{(B.8)}$$

$$y_2=l_1\sin(\theta_1)+\tfrac{1}{2}l_2\sin(\theta_1+\theta_2)\;\;\dot{y}_2=-l_1\dot{\theta}_1\cos(\theta_1)-\tfrac{1}{2}l_2(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2)\quad\text{(B.9)}$$

Then the velocity of the mass center of the link 2 is calculated.

$$v_2^2=l_1^2\dot{\theta}_1^2+\tfrac{1}{4}l_2^2(\dot{\theta}_1+\dot{\theta}_2)^2+l_1l_2(\dot{\theta}_1^2+\dot{\theta}_1\dot{\theta}_2)\cos(\theta_2)\quad\text{(B.10)}$$

The Lagrangian is given in Equation B.11.

$$\begin{aligned}L=&\tfrac{1}{8}m_1l_1^2\dot{\theta}_1^2+\tfrac{1}{2}I_1\dot{\theta}_1^2+\tfrac{1}{2}m_2l_1^2\dot{\theta}_1^2+\tfrac{1}{8}m_2l_2^2(\dot{\theta}_1+\dot{\theta}_2)^2\\&+\tfrac{1}{2}m_2l_1l_2(\dot{\theta}_1^2+\dot{\theta}_1\dot{\theta}_2)\cos(\theta_2)+\tfrac{1}{2}I_2(\dot{\theta}_1+\dot{\theta}_2)^2\end{aligned}\quad\text{(B.11)}$$

Taking the derivative of the Lagrangian gives you the two equations of motion in Equations B.12 and B.13.

$$\begin{aligned}&\ddot{\theta}_1[\tfrac{1}{4}m_1l_1^2+I_1+m_2(l_1^2+\tfrac{1}{4}l_2^2+l_1l_2\cos(\theta_2))+I_2]+\ddot{\theta}_2[m_2(\tfrac{1}{4}l_2^2+\tfrac{1}{2}l_1l_2\cos(\theta_2))+I_2]\\&-\tfrac{1}{2}m_2l_1l_2\sin(\theta_2)(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2)=T_1\end{aligned}\quad\text{(B.12)}$$

$$\ddot{\theta}_1[m_2(\tfrac{1}{4}l_2^2+\tfrac{1}{2}l_1l_2\cos(\theta_2))+I_2]+\ddot{\theta}_2[\tfrac{1}{4}m_2l_2^2+I_2]+\tfrac{1}{2}m_2l_1l_2\dot{\theta}_1^2\sin(\theta_2)=T_2\quad\text{(B.13)}$$

After combining Equations B.12 and B.13 into a matrix representation, you get Equation B.14.

$$\begin{bmatrix} \frac{1}{4}m_1 l_1^2 + I_1 + m_2(l_1^2 + \frac{1}{4}l_2^2 + l_1 l_2 \cos(\theta_2)) + I_2 & m_2(\frac{1}{4}l_2^2 + \frac{1}{2}l_1 l_2 \cos(\theta_2)) + I_2 \\ m_2(\frac{1}{4}l_2^2 + \frac{1}{2}l_1 l_2 \cos(\theta_2)) + I_2 & \frac{1}{4}m_2 l_2^2 + I_2 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} =$$
$$\begin{Bmatrix} T_1 \\ T_2 \end{Bmatrix} + \begin{bmatrix} 0 & \frac{1}{2}m_2 l_1 l_2 \sin(\theta_2) \\ -\frac{1}{2}m_2 l_1 l_2 \sin(\theta_2) & 0 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{Bmatrix} + \begin{bmatrix} m_2 l_1 l_2 \sin(\theta_2) \\ 0 \end{bmatrix} \left\{ \dot{\theta}_1 \dot{\theta}_2 \right\} \tag{B.14}$$

To get the complete equations of motion, you must combine Equations B.7 and B.14 to get Equation B.15,

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} = \begin{Bmatrix} T_1 \\ T_2 \end{Bmatrix} + \begin{bmatrix} 0 & c \\ -c & 0 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{Bmatrix} + \begin{bmatrix} 2c \\ 0 \end{bmatrix} \left\{ \dot{\theta}_1 \dot{\theta}_2 \right\} \tag{B.15}$$

where

$$\begin{aligned} M_{11} &= (m_e + m_p + \tfrac{1}{4}m_1 + m_2)l_1^2 + (m_p + \tfrac{1}{4}m_2)l_2^2 + (2m_p + m_2)l_1 l_2 \cos(\theta_2) + I_1 + I_2 + I_e + I_p \\ M_{12} = M_{21} &= (m_p + \tfrac{1}{4}m_2)l_2^2 + (m_p + \tfrac{1}{2}m_2)l_1 l_2 \cos(\theta_2) + I_2 + I_p \\ M_{22} &= (m_p + \tfrac{1}{4}m_2)l_2^2 + I_2 + I_p \\ c &= (m_p + \tfrac{1}{2}m_2)l_1 l_2 \sin(\theta_2) \end{aligned} \tag{B.16}$$

After inverting the mass matrix and multiplying both sides by the inverse, Equation B.17 is the result. This is the form that the equations were entered into MATLAB's nonlinear integrator.

$$\begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} = \frac{1}{M_{11}M_{22} - M_{12}^2} \begin{Bmatrix} M_{22}(T_1 + c\dot{\theta}_2^2 + 2c\dot{\theta}_1\dot{\theta}_2) - M_{12}(T_2 - c\dot{\theta}_1^2) \\ -M_{12}(T_1 + c\dot{\theta}_2^2 + 2c\dot{\theta}_1\dot{\theta}_2) + M_{11}(T_2 - c\dot{\theta}_1^2) \end{Bmatrix} \tag{B.17}$$

Torque T_1 is the driving torques acting between the base and shoulder pitch joint. Torque T_2 is the driving torque acting at the elbow joint between links 1 and 2. Proportional derivative controller loops were closed around the joint angles to get the torques. The stiffnesses and damping ratios, which could also be considered the proportional and derivative controller gains, are given in Chapter 5.2.1.

$$T_1 = K_1(\theta_{1ref} - \theta_1) + B_1(\dot{\theta}_{1ref} - \dot{\theta}_1) \tag{B.18}$$

$$T_2 = K_2(\theta_{2ref} - \theta_2) + B_2(\dot{\theta}_{2ref} - \dot{\theta}_2) \tag{B.19}$$

K_1, K_2, B_1, and B_2 can be adjusted to give the system different frequencies and damping ratios.

Using the DRS Simulation

Appendix C

The DRS model is surprisingly easy to use, if you have the proper documentation. The two documents needed are *The DRS Users Guide* and the *Payload Deployment and Retrieval System Simulation Database.* [16, 19] They contain the default parameter values and a description of most of the initialization variables. The DRS has been converted to FORTRAN code that can be compiled on a UNIX machine or a Macintosh. The DRS program consists of 13 separate FORTRAN files. Usually, the only programs that have to be modified are vname-orig.f, vdatout.f, and vrmsgpc_inp.f.

vname-orig.f contains all of the initialization parameters. This is where you can change the payload, the friction and stiction levels, the integral gain, the initial joint angles and velocities, and many others. After the following line you can change or insert any variables in Appendix C of *The DRS User's Guide.*

```
C ** BEGIN NAMELIST DECLARATION
```

If you want to see what the default variables are, comment out all of the variables that follow that line, compile, and run the program. The first 435-437 lines of the output file will contain the initial values of the simulation parameters. One thing I found useful to do was to let the program read in the initial joint angles. Otherwise you have to recompile the program every time you change initial angles or rates. For example, if you wanted to read in the initial shoulder pitch and elbow pitch angles you would add the following code to the end of vname-orig.f.

```
open(unit=2,status='old',file='intheti.dat')
read(2,*)gam(3),gam(4)
```

Just make sure that you actually have a file called 'intheti.dat', or the program will crash. This file can be created in MATLAB very easily, as the following line shows. The initial angles are in degrees.

```
save intheti.dat -45 45 -ascii
```

To change the output variables, you must edit vdatout.f. Search for the phrase 'write(6' and you will find something close to the following line. Chose whatever variables you want from Appendix B of *The DRS User's Guide* and insert them into this line. Right now this line prints out the time, shoulder pitch angle, elbow pitch angle, x tip position, y tip position, and z tip position. Do not output too many variables, unless you plan on creating 1MB files every time you run the DRS.

```
2009 WRITE(6,2010) TYME,XDATA(23),XDATA(24),XDATA(40),
    1XDATA(41),XDATA(42)
```

In order to change the inputs to the program, you must edit vrmsgpc_inp.f. Search for the phrase 'read(3' and you will find the following line. *jtcdfp* is the command to the motor. The following lines are setting up the input file and reading in shoulder pitch and elbow pitch rates from the file 'indat.dat'.

```
c    nsing update
c    arbitrary velocity input
           integer numb,jnt
           character *100 outdat, indat
           COMMON /OFILE/outdat,indat
c   nsing update
C   override the SRMS gpc command and put mine in.
       inquire(file=indat,number=numb)
       if (numb .eq. 3) goto 90
       open(unit=3,file=indat)
 90    continue
       do 100 i=1,6
          jtcdfp(i)=0.0
 100   continue
C    shoulder yaw is 1
C    shoulder pitch is 2
C    elbow pitch is 3
       read(3,*)jtcdfp(2),jtcdfp(3)
```

In MATLAB, you need to calculate the joint rates you wish to input, in rad/sec. The joints must be multiplied by gains to convert them to motor commands. Each joint rate must be multiplied by the gear ratio and the arm command scaling factor K_1=12.0948. The gear ratios are as follows:

Shoulder yaw	Shoulder pitch	Elbow pitch	Wrist pitch	Wrist yaw	Wrist roll
1841.95	1842.95	1260.28	737.74	738.74	737.74

Table C.1: DRS joint gear ratios

The inputs to the DRS must be integers, so you must round the input after multiplying it by the conversion factors. If you were creating the input data file in MATLAB, you would do the following:

```
Kelb=1260*12.0948;Ksh=1843*12.0948;
dthet=round([dShP*Ksh dElP*Kelb]);
save indat.dat  dthet -ascii
```

Use the following Makefile to compile the program.

```
.SUFFIXES: .f .o

ORIG=drsmain.o vdatout.o vflex1.o vflex2.o vinterps.o vonetym.o
vphm.o vrunmod.o vupdate.o vinit.o vservo.o vname-orig.o
vrmsgpc_inp.o

orig: ${ORIG}
      f77    -o drsm-orig ${ORIG} libimsld.a
      @echo "done"

.f.o: $*.f
      f77   -c $*.f

.c.o: $*.c
      cc   -g -c $*.c
```

After the program is compiled, the executable file is `drsm-orig`. Rename the file and run as follows:

```
drsm-orig >datafile.dat
```

Of course, you can choose the name of the output file to be anything you want. To look at the data, edit the output file by removing the first 435-437 lines of text. The text contains all of the namelist initialization parameters that the DRS uses. Then load in the file to MATLAB and you should be able to look at the data you have created. The DRS also creates a file called outdat.dat, which should be deleted as it contains garbage, and grows very large.

If you want to change the payload, you must change the payload mass, inertia, position vectors, and joint rate limits. The *Payload Deployment and Retrieval System Simulation Database* gives a good description of payload parameters for certain payloads.