RESIDUAL VIBRATION REDUCTION ON A THREE MODE FLEXIBLE SYSTEM USING INPUT SHAPING

By

IVAN LEONEL BLANCO

Bachelor Of Science
Oklahoma State University
Stillwater, Oklahoma
1999

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
August, 2001

RESIDUAL VIBRATION REDUCTION ON A THREE

MODE FLEXIBLE SYSTEM USING

INPUT SHAPING

Thesis Approved:

Thesis Adviser

Gary E. Young

Dr. Prabhakar Pagilla

Alfred Carlozzi

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my gratitude to my graduate advisor, Dr. Eduardo A. Misawa, for his constructive guidance, instruction, and supervision.

I would like to acknowledge and thank my masters committee members: Dr. Prabhakar R. Pagilla and Dr. Gary E. Young for their support and suggestions in completion of this research.

I would also like to thank my present and former colleagues at Oklahoma State University Ban Fu Chee, Brian O'Dell, Todd Lyle, Thomas Rendon, Chad Stoecker and Hanz Richter.

On a more personal note, I would like to express my appreciation to my mother and father for giving me the inspiration for pursuing an advanced degree. Also my fiancee, Carrie, has helped me in my undergraduate and graduate work with her support and patience.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

Figure **Page**

Figure **Page**

NOMENCLATURE

J_1	Inertia of First Disk
J_2	Inertia of Second Disk
J_3	Inertia of Third Disk
c_1	Damping Ratio at Disk One
c_2	Damping Ratio at Disk Two
c_3	Damping Ratio at Disk Three
k_1	Spring Constant at Disk One
k_2	Spring Constant at Disk Two
k_3	Spring Constant at Disk Three
θ_1	Angular Position of Disk One
θ_2	Angular Position of Disk Two
θ_3	Angular Position of Disk Three
$\dot{\theta}_1$	Angular Velocity of Disk One
$\dot{\theta}_2$	Angular Velocity of Disk Two
$\dot{\theta}_3$	Angular Velocity of Disk Three
$\ddot{\theta}_1$	Angular Acceleration of Disk One
$\ddot{\theta}_2$	Angular Acceleration of Disk Two
$\ddot{\theta}_3$	Angular Acceleration of Disk Three
u	System Input
x	State Vector$(\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)$
$\dot{x} = Ax + Bu$	System State Space Model
$y = Cx + Du$	System Output

ω_i	Natural Frequncy of the i^{th} Mode
ζ_i	Damping Ratio of the i^{th} Mode
ω_{d_i}	Damped Natural Frequency of the ith Mode, $\omega_{d_i} = \omega_i\sqrt{1-\zeta_i^2}$
A_{ij}	j^{th} Impulse Amplitude for the i^{th} Mode
t_j	Time of the j^{th} Impulse
t_N	Time of the Last Impulse
p_i	Shaper Pole
p_i^*	Complex Conjugate of Shaper Pole
Subscripts :	
i	i^{th} Mode
j	j^{th} Impulse
N	Total Number of Impulses
Superscripts :	
$*$	Complex Conjugate
T	Transpose

Chapter 1

Introduction

This chapter is devoted to presenting the thesis format, describing the model used in simulation and experimentation, and describing the motivation for this research project. This paper presents a few techniques that are commonly used on flexible structures today. It also introduces a new technique that is based on the work by N.C Singer.

1.1 Motivation for Research

Exciting the natural resonances of a flexible system can cause excessive wear, noise, failure, or poor performance. Therefore, many researchers have devoted their resources to finding methods to reduce or completely eliminate unwanted vibration. Adding stiffness to flexible structures is one obvious solution to excess vibration, but many structures major design criteria is weight reduction. The Space Shuttle remote manipulator is a good example of a

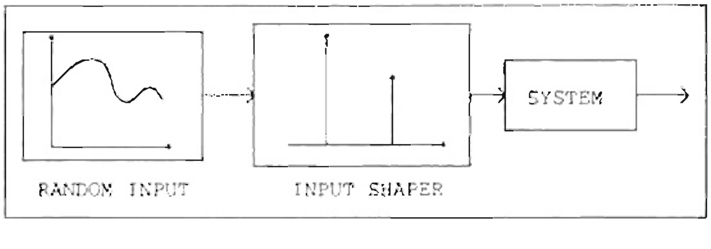

Figure 1.1: Input Shaping

structure that must be light weight and perform precise movement. It is actually the model used in Singer's Ph.D. thesis [12], who was the first person to derive robustness constraints for input shaping. Ideally researchers would like to be able to completely eliminate residual vibration with control strategy instead of adding stiffness.

1.2 Thesis Structure Description

Chapter 2 briefly reviews some techniques used to reduce residual vibration and detailed descriptions of a few input shaping techniques. Chapter 3 derives the mathematical model of the mechanical system used in simulations and experiments. It also provides a brief description of the hardware and software used in the experiments. Chapter 4 describes detailed derivation of input shapers developed by Singer and an extension of his work. Chapter 5 details the derivation of input shapers developed using Tuttle's technique in the discrete Z-Domain. Chapter 6 describes the method to derive more robust input shapers and "Time-Optimal" input shapers, techniques developed by Singhose. Chapters 4, 5, and 6 all provide simulation and experimental results obtained using the model outlined in chapter 3. Chapter 7 describes how these techniques were implemented on the hardware and presents any problems with applying the techniques. Comparisons on the performance of the input shapers are made in chapter 8. Final comments, conclusions, contributions, and topics for future work are described in chapter 9. The MATLAB files used in simulation to test the input shapers and create trajectory files for experiments are shown in appendix B, C, D, and E.

Chapter 2

Literature Review

Input Shaping can be defined as convolving an arbitrary command signal with an input shaper, and using the result as the new system input. The shaper is a sequence of positive or positive and negative impulses with finite time. The shaped command signal should execute commands without exciting a system's resonant frequency or frequencies. Figure 2.1 illustrates the input shaping technique for an open loop system, but input shaping is not limited to open loop systems. It can also be implemented on closed loop systems.

The original pioneer of input shaping was Otto Smith [34] in the late 1950's. Smith developed a technique called Posicast control that split a unit step input into two steps that sum to one. The second step in Posicast control cancels the vibration that the first step excites. Therefore, a system could be moved without vibration. The drawback to Posicast control is that it is not robust to modeling errors and it only works if the system parameters are exact. Singer [12] developed constraint equations to improve the robustness of input

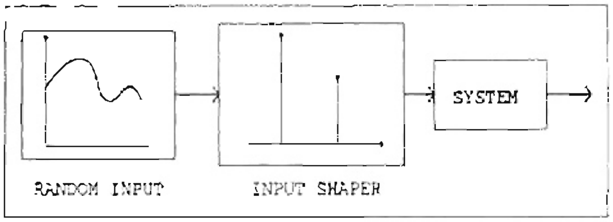

Figure 2.1: Input Shaping Structure

shaping and accommodate systems with multiple modes. The level of necessary robustness is determined by the user. The time lag that the input shaper introduces increases with robustness. Smith's and Singer's input shaping technique are both derived and implemented in the continuous time, but several researchers have devoted their efforts to frequency domain input shaping approaches [1, 19, 35].

Singh and Vadali [19] used time delays in the Laplace domain to cancel the system poles and found that time delays are similar to Singer's input shaping technique. Bhat and Miu [1] found a set of constraint equations based on point-to-point control in the continuous time and simplified them by finding the Laplace transform of the constraints. Tuttle and Seering [35] used a discrete domain zero placement technique to cancel the system poles for multiple mode systems. The primary drawback of some of these techniques is that they introduce a time lag into the rise time of a response equal to the overall length of input shaper. Singhose et. al. [30, 32, 33] has developed several "time-optimal" input shapers that minimize the duration of input shapers but require negative impulses. Singhose [21, 22, 26, 28] also used Singer's vector diagram representation to widen the region of insensitivity of an input shaper. Though there are many input shaping and command shaping techniques available today, this research project focuses on techniques developed by Singer, Singhose, and Tuttle.

2.1 Singer's Preshaping Command Technique

Neil Singer's preshaping command technique is based on generating a vibration free input [12, 15]. Singer accomplished this first by defining an expression for a system response to an impulse input. The basis for choosing an expression was that any linear flexible system could be specified as a set of cascaded second-order poles with a decaying sinusoidal response to an impulse input, described as

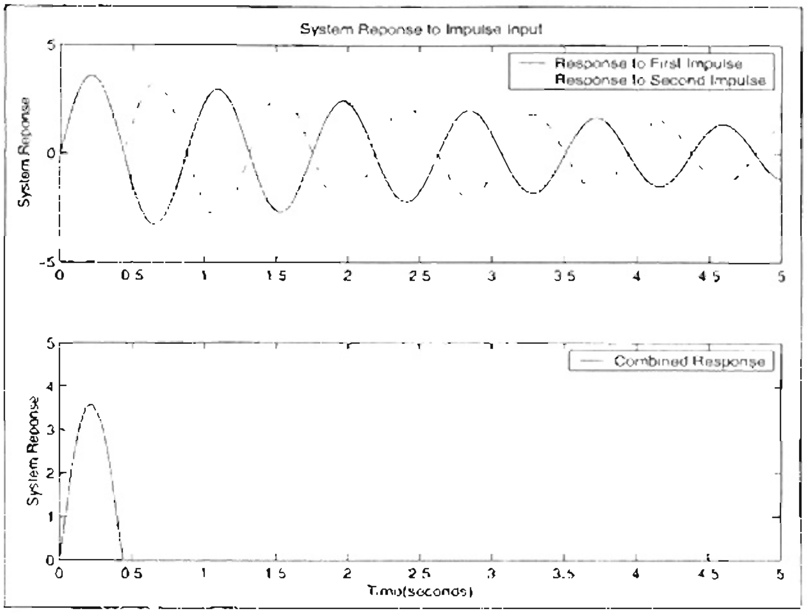

Figure 2.2: System Response To Single Impulse And To Two Impulses Combined

$$y(t) = \left[A \frac{\omega_0}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_0(t-t_0)} \right] sin(\omega_0 \sqrt{1-\zeta^2}(t-t_0)) \tag{2.1}$$

A is the impulse amplitude, ω is the natural frequency of a mode, ζ is the damping ratio of a mode, t_0 is the time of the impulse, and t is time. Singer assumes that numerator dynamics are not present in the plant. The justification for this assumption can be found in [12].

Now assume that only one mode is present in a system. The design objective is to give a system a series of impulses in a manner such that the system does not vibrate after the time of the last impulse is applied. A two impulse sequence is the shortest input shaper and will be used in the derivation of the constraint equations. Figure 2.2 is generated by equation 2.1 for a single mode system over an arbitrary time after the time of each individual impulse. Figure 2.2 illustrates a system response to a single impulse and the system response to both impulses combined. It shows that after the second impulse the system response does not vibrate.

2.1.1 Derivation of Time Domain Input Shaper

In order to find an expression for the amplitude of vibration, A_{amp}, a trigonometric identity was implemented to add the response of the two decaying sinusoidal responses:

$$B_1 \sin(\alpha t + \phi_1) + B_2 \sin(\alpha t + \phi_2) = A_{amp} \sin(\alpha t + \Psi) \tag{2.2}$$

where,

$$A_{amp} = \sqrt{(B_1 \cos\phi_1 + B_2 \cos\phi_2)^2 + (B_1 \sin\phi_1 + B_2 \sin\phi_2)^2} \tag{2.3}$$

$$\psi = \tan^{-1}\left(\frac{B_1 \cos\phi_1 + B_2 \cos\phi_2}{B_1 \sin\phi_1 + B_2 \sin\phi_2}\right) \tag{2.4}$$

Using the result from equation 2.2 an expression for the Amplitude of Vibration for a Multi-Impulse Input was derived:

$$A_{amp} = \sqrt{\left(\sum_{j=1}^{N} B_j \cos\phi_j\right)^2 + \left(\sum_{j=1}^{N} B_j \sin\phi_j\right)^2} \tag{2.5}$$

where, $B_j = \frac{A_j \omega}{\sqrt{1-\zeta^2}} e^{-\zeta\omega(t_N - t_j)}$ and $\phi_j = t_j \omega \sqrt{1-\zeta^2}$

B_j is the coefficient of the sine term in equation 2.1 for the j^{th} impulse at time, t_j, where t_N is the time of the last impulse. Note that B_j is specified for a single mode system, where ζ and ω are the damping ratio and undamped natural frequency of the mode, respectively. To eliminate residual vibration in a flexible system it is required that the amplitude of vibration, A_{amp}, be zero after the time of the last impulse. This is accomplished by forcing the individual squared sums in equation 2.5 be independently zero. That is,

$$B_1 \cos\phi_1 + B_2 \cos\phi_2 + \ldots + B_N \cos\phi_N = 0 \tag{2.6}$$

A_j	Amplitude of the j^{th} Impulse
t_j	Time of the j^{th} Impulse
ζ_i	Damping Ratio of the i^{th} Mode
ω_i	Natural Frequency of the i^{th} Mode
t_N	Sequence End Time

Table 2.1: Variables Definitions

$$B_1 \sin\phi_1 + B_2 \sin\phi_2 + \ldots + B_N \sin\phi_N = 0 \tag{2.7}$$

From equations 2.6 and 2.7 a more general form of the zero vibration(ZV) constraint was found to be:

$$\sum_{j=1}^{N} A_j e^{-\zeta_i\omega_i(t_N - t_j)} \sin(t_j\omega_i\sqrt{1-\zeta_i^2}) = 0 \tag{2.8}$$

$$\sum_{j=1}^{N} A_j e^{-\zeta_i\omega_i(t_N - t_j)} \cos(t_j\omega_i\sqrt{1-\zeta_i^2}) = 0 \tag{2.9}$$

For the given constraint equations, consider the case when $N = 2$. This yields the following impulse amplitudes and impulse times: A_1, A_2, t_1, and t_2. It is common practice in input shaping to assume that the first impulse, A_1, is equal to one and its impulse time is $t_1 = 0$. Now it is easy to see that there are two unknowns, A_2 & t_2, and two constraint equations, 2.8 and 2.9.

There are an infinity number of solutions to the constraint equations because of their transcendental nature. Therefore, Singer applied additional constraints to determine a solution for A_2 and t_2. He required that all the impulse amplitudes be positive and that the time-duration of the shaper be the shortest possible that satisfies the aforementioned constraint. In [12] Singer found the amplitude of the second impulse to be:

$$A_2 = K \tag{2.10}$$

at,

$$t_2 = \Delta T \tag{2.11}$$

where K and ΔT will be defined later.

In order to insure that the un-shaped command signal and the shaped command signal provide the same steady state response, Singer required that all the impulse amplitudes sum to one. That is equivalent to the following constraint:

$$\sum_{j}^{N} A_j = 1 \tag{2.12}$$

Therefore, for the case when $N = 2$ the solution for the impulse amplitudes, A_1 and A_2, becomes:

$$A_1 = \frac{1}{1+K} \tag{2.13}$$

$$A_2 = \frac{K}{1+K} \tag{2.14}$$

where,

$$K = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{2.15}$$

$$\Delta T = \frac{\pi}{\omega\sqrt{1-\zeta^2}} \tag{2.16}$$

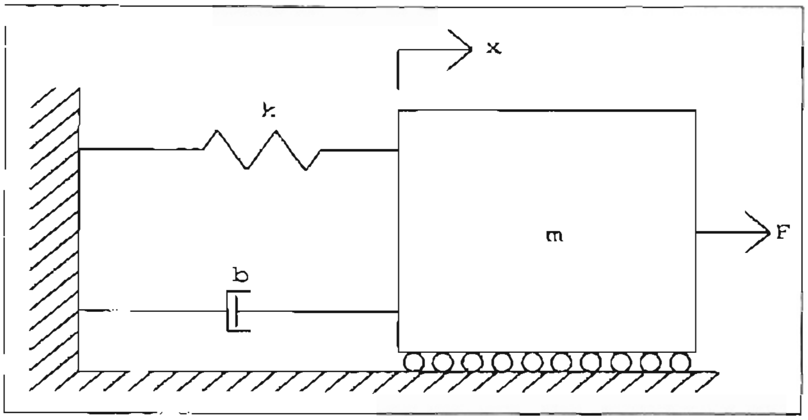

Figure 2.3: Mass Spring Damper

Note that ΔT is the time of the first overshoot(one half of the period of damped oscillation) and K is the step response overshoot of a 2-pole linear system with no numerator dynamics. For example, the mass spring damper shown in figure 2.3 is a 2-pole linear system. The transfer function for the system is defined as,

$$T(s) = \frac{1/m}{s^2 + \frac{b}{m}s + \frac{k}{m}} \tag{2.17}$$

Define the system parameters as $m = 1$, $k = 1$, and $b = 0.5$. The transfer function 2.17 can be redefined to find the system's natural frequency(ω_n) and damping ratio(ζ).

$$\therefore T(s) = \frac{1}{s^2 + 0.5s + 1} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.18}$$

Using equation 2.18, the system's modal properties were calculated to be $\omega_n = 1$ rad/sec and $\zeta = 0.25$. The overshoot level and time was calculated with equation 2.15 and 2.16. The calculated results were verified by plotting the unit step response of the system shown in figure 2.4.

Figure 2.5 illustrates the shaper found on equations 2.13 and 2.14. The shaper found is referred to as a first-order with respect to robustness to modeling errors in frequency and in damping. This means that the shaper can completely eliminate vibration from an arbitrary input only if the system properties(ζ & ω) are known exactly.

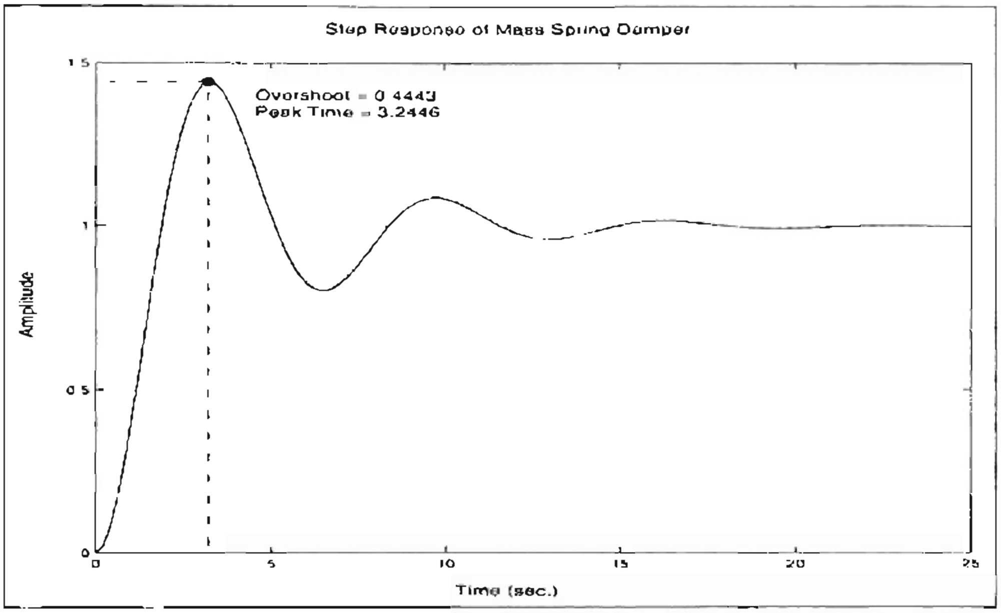

Figure 2.4: Mass Spring Damper Step Response

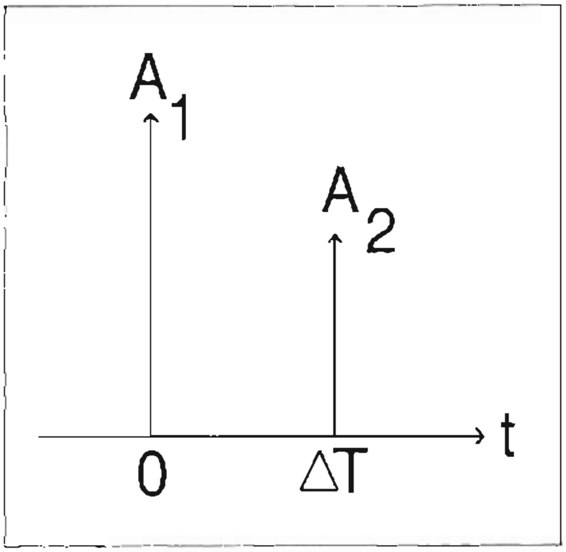

Figure 2.5: Two-Impulse Input Shaper

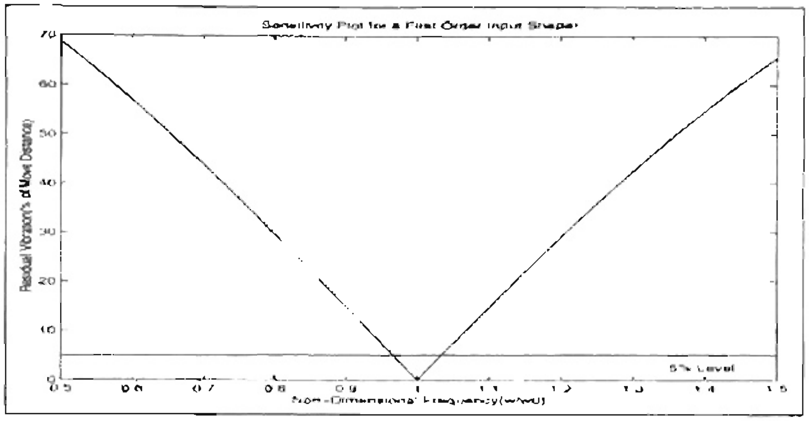

Figure 2.6: First Order Shaper Sensitivity

To illustrate the sensitivity of a first order input shaper a ratio of the amplitude of vibration, A_{amp}, with input shaping to the amplitude of vibration without input shaping can be formed using equation 2.5 where $N = 2$.

$$V = e^{-\zeta\omega t_N}\sqrt{\left(\sum_{j=1}^{N} A_j e^{\zeta\omega t_j}\sin(t_j\omega\sqrt{1-\zeta^2})\right)^2 + \left(\sum_{j=1}^{N} A_j e^{\zeta\omega t_j}\cos(t_j\omega\sqrt{1-\zeta^2})\right)^2} \tag{2.19}$$

By varying the frequency, ω, from the system's nominal natural frequency, ω_0, figure 2.6 illustrates how sensitive a first order input shaper is to changes in natural frequency. The horizontal axis is a non-dimensional frequency scale (ω/ω_0). A line at 5% residual vibration is shown to illustrate the bound for an acceptable response. Note that when $\omega/\omega_0 = 1$ a first order input shaper can completely eliminate all unwanted vibration, but the slope of the curve is very sharp around the nominal natural frequency making small errors in natural frequency significant. A measure of the level of robustness of a first order shaper is to consider the width of figure 2.6, at the 5% level of acceptable vibration. Therefore, the level of robustness for a first order shaper is less than ≈ ±5% variation in natural frequency. Singer [12, 15] shows that variations in the damping ratio does not effect the level of residual vibration significantly.

To increase the level of robustness, Singer derived two additional constraint equations by taking the derivative of original constraint equations 2.8 and 2.9 with respect to the natural frequency, ω. This reduces the amount of residual vibration induced by small changes in frequency. The result of taking the derivatives of equations 2.8 and 2.9 is,

$$\sum_{j=1}^{N} A_j t_j e^{-\zeta_i \omega_i (t_N - t_j)} \sin(t_j \omega_i \sqrt{1-\zeta_i}) = 0 \qquad (2.20)$$

$$\sum_{j=1}^{N} A_j t_j e^{-\zeta_i \omega_i (t_N - t_j)} \cos(t_j \omega_i \sqrt{1-\zeta_i}) = 0 \qquad (2.21)$$

One more impulse is included with the addition of two constraint equations. Therefore, $N = 3$ and the impulse amplitudes and impulse times are: A_1, A_2, A_3, t_1, t_2, and t_3. This sequence of three impulses is called a second order input shaper and again the first impulse, A_1, is assumed to equal to one at $t_1 = 0$. The solution for the three impulse sequence is as follows.

$$A_1 = \frac{1}{1 + 2K + K^2} \quad @ \quad t_1 = 0 \qquad (2.22)$$

$$A_2 = \frac{2K}{1 + 2K + K^2} \quad @ \quad t_2 = \Delta T \qquad (2.23)$$

$$A_3 = \frac{K^2}{1 + 2K + K^2} \quad @ \quad t_3 = 2\Delta T \qquad (2.24)$$

It can be seen from the denominator of A_1, A_2, and A_3 that the impulse amplitudes have been normalized according to equation 2.12. Also, the time of the last impulse, t_3, is two times the damped period of oscillation. Therefore, the time duration of a second order input shaper is two times the duration of a first order input shaper. Despite the longer time duration of a second order input shaper, figure 2.7 shows the level of robustness gained

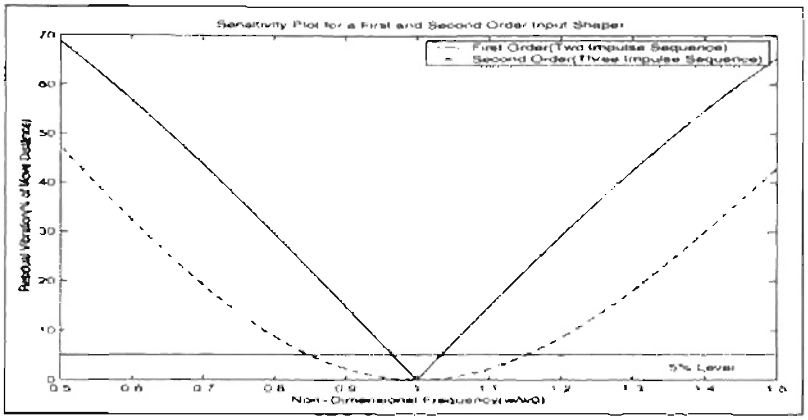

Figure 2.7: Second Order Shaper Sensitivity

by adding another impulse. The level of robustness for a second order input shaper is less than ≈ ±20% variation in natural frequency. Also, Singer [12] showed that the derivative constraints also provide the same level of robustness for variations in damping, ζ.

If a controls engineer has the need for an even more robust input shaper, then he may use the general form for the derivative of the initial constraint equations 2.8 and 2.9. The expressions for the q^{th} derivative of the initial constraints are:

$$\sum_{j=1}^{N} A_j(t_j)^q e^{-\zeta\omega(t_N - t_j)} sin(t_j \omega \sqrt{1-\zeta^2}) = 0 \tag{2.25}$$

$$\sum_{j=1}^{N} A_j(t_j)^q e^{-\zeta\omega(t_N - t_j)} cos(t_j \omega \sqrt{1-\zeta^2}) = 0 \tag{2.26}$$

For every two constraint equations added, one impulse is added to the sequence of impulses. This makes the impulse sequence more robust to modeling errors, but increases the time-duration of the shaper. Consider the case when $q = 2$, this means that the second derivatives of the original constraints are derived. Therefore, the impulse sequence becomes: A_1, A_2, A_3, A_4, t_1, t_2, t_3, and t_4. The impulse amplitudes and times are:

$$A_1 = \frac{1}{1 + 3K + 3K^2 + K^3} \quad @ \quad t_1 = 0 \tag{2.27}$$

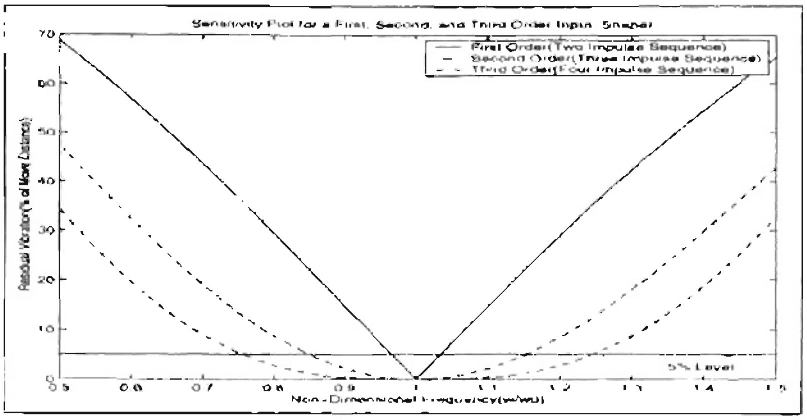

Figure 2.8: Third Order Shaper Sensitivity

$$A_2 = \frac{3K}{1 + 3K + 3K^2 + K^3} \quad @ \quad t_2 = \Delta T \tag{2.28}$$

$$A_3 = \frac{3K^2}{1 + 3K + 3K^2 + K^3} \quad @ \quad t_3 = 2\Delta T \tag{2.29}$$

$$A_4 = \frac{K^3}{1 + 3K + 3K^2 + K^3} \quad @ \quad t_4 = 3\Delta T \tag{2.30}$$

The time duration of a third order input shaper is 1.5 times the period of damped oscillation. By now it is easy to see that the times for each successive impulse is an equally spaced interval based on integer multiples of half a period of damped oscillation. Again, the drawback of a third order input shaper is that it is longer than a first or second order input shaper, but figure 2.8 shows that the third order shaper is more robust than the two previous examples.

A third order shaper is approximately -30 % to +40 % robust to modeling errors. Table 2.2 compares the level of robustness to modeling errors and time durations for the three input shapers derived.

Shaper	Level of Robustness to Model Variations	Time Duration(Order$*\Delta T$)
First Order	$\approx \pm 5\%$	1
Second Order	$\approx \pm 20\%$	2
Third Order	$\approx$ -30 % to +40%	3

Table 2.2: Comparison of Sensitivity for First, Second, and Third Order Shapers

2.1.2 Command Shaping for Systems with Multiple Modes

An input shaper designed for a single mode can excite un-modeled dynamics resulting in some level of residual vibration. With Singer's command shaping technique there are a couple of ways to minimize residual vibration in multiple mode systems. Shapers can be designed for each mode in the system with different or the same level of robustness(order). Then, to find the input shaper that compensates for all the modes the individual shapers are convolved. The time duration of the multiple mode input shaper is the sum of the time durations for each individual shaper. This method is sometimes called the convolved solution for a multiple mode input shaper. Another method to finding a multiple mode shaper is to solve for shaper directly by using the constraint equations. To design a first order shaper for a system with two modes, the shaper would have three impulses. This method is appropriately defined as the direct solution. A nonlinear optimization package is required to find the direct solution. The direct solution always contains fewer impulses than the convolved solution. Therefore, the direct solution is always faster than the convolved solution.

2.2 Tuttle's Zero-Placement Technique

This technique was researched by T.D. Tuttle[35] and it has be found to be useful because of its ease of application to arbitrary inputs. This technique is also based on the work of O.J. Smith [34]. Smith showed that Posicast commands, when converted to the Laplace S-domain or discrete Z-domain, cancelled the system poles. The drawback of Posicast control is that it is sensitive to variations in model parameters. N.C. Singer [12] presented a technique to improve the robustness of input shapers. Section 2.1.1 shows the derivation and properties of Singer's time domain input shaping technique. It has been shown that adding additional shaper zeros at the system poles provides an equivalent level of robustness to Singer's technique.

2.2.1 Derivation of Zero-Placement Multiple Mode Input Shaper

Since traditional input shapers are only defined at discrete time intervals, a discrete frequency domain representation is suitable because it too is defined at discrete intervals. Tuttle's technique consist of deriving a shaper in the discrete domain, mapping it to the continuous Laplace domain, and then using the inverse Laplace transform to find the input shaping sequence in the time domain. Five conditions are defined to derive the general form of a zero-placement input shaper. These conditions are translated into mathematical discrete domain constraints.

The conditions are: 1. The shaper must eliminate vibration at all unwanted modes. 2. The shaper must provide adequate robustness to uncertainty in model parameters(ζ_i & ω_i). 3. The shaper must be causal. 4. The shaper must minimize distortion of the input command. 5. The shaped command must not violate actuator limits. These conditions are based on achieving maximum performance while minimizing unwanted residual vibration.

The discrete frequency domain equivalents of the five requirements of zero-placement technique are outlined below:

1. The Shaper Must Eliminate Vibration at All Unwanted Modes

   This places an input shaper zero at each system pole. For flexible systems, the system poles are complex conjugate pairs and therefore the input shaper zeros are also complex conjugate pairs. The zeros are defined for the i^{th} mode as p_i and p_i^*, where

p_i^* is the complex conjugate of p_i.

$$p_i = e^{-\zeta_i \omega_{ni} T} e^{j\omega_{di} T} \tag{2.31}$$

$$p_i^* = e^{-\zeta_i \omega_{ni} T} e^{-j\omega_{di} T} \tag{2.32}$$

where ω_{di} is defined as the damped natural frequency of the i^{th} mode,

$$\omega_{di} = \omega_{ni}\sqrt{1 - \zeta_i^2} \tag{2.33}$$

The initial input shaper that satisfies this constraint is defined as,

$$H(z) = (z - p_1)(z - p_1^*)(z - p_2)(z - p_2^*)\ldots(z - p_m)(z - p_m^*) \tag{2.34}$$

where a system with m unwanted modes of vibration, the shaper must contain $2m$ zeros to cancel the system poles.

2. The Shaper Must have Adequate Robustness to Parameter Uncertainty

   The derivative constraint used by Singer to improve robustness to model uncertainty can be implemented in the discrete frequency domain by adding additional shaper zeros at the system poles.

   The shaper designed to increase robustness is defined as,

$$H(z) = (z - p_1)^{n_1}(z - p_1^*)^{n_1}\ldots(z - p_m)^{n_m}(z - p_m^*)^{n_m} \tag{2.35}$$

   if $n_1 = 2$ then the input shaper has second-order robustness to errors in system parameters for the first mode in a multiple mode system. This is equivalent to adding two additional constraint equations for the first mode.

3. The Shaper Must be Causal

   To ensure that the shaper remains causal or non-anticipative, the number of shaper poles must be greater than the number of shaper zeros. Therefore, the denominator of $H(z)$ must be higher order than the numerator.

4. The Shaper Must Minimize Distortion of the Input Command

   Since input shaping introduces a time lag into the system input equal to the length of the shaper, the number of poles and zeros should be minimized while still satisfying robustness constraints. To meet this constraint and still satisfy the causality constraint the input shaper should have an equal number of poles and zeros. Also to eliminate denominator dynamics from the input shaper, all of the shaper poles should be at the z-plane origin.

   The equation that satisfies the constraints is:

$$H(z) = \frac{C(z-p_1)^{n_1}(z-p_1^*)^{n_1}\dots(z-p_m)^{n_m}(z-p_m^*)^{n_m}}{z^r} \tag{2.36}$$

   where, $r = 2(n_1 + \cdots + n_m)$ and C is constant that will be defined by the next constraint.

5. The Shaped Command Must Not Violate Actuator Limits

   Singer satisfied this constraint by requiring a shaper with positive impulse amplitudes that sum to one. The resulting shaper will not violate actuator constraints given that the un-shaped command input does not exceed actuator constraints. Tuttle uses two additional steps to find an impulse sequence in the continuous time. The steps are:

   - Map the Discrete Transfer Function to the S-Plane by Using $z = e^{sT}$
   - Take the Inverse Laplace Transform to Find the Impulse Sequence

     Now the impulse sequence can be represented in continuous time as,

$$h(t) = C'[\delta(t) + a_1\delta(t-T) + a_2\delta(t-rT) + \dots + a_r\delta(t-rT)] \tag{2.37}$$

     where, C' is Defined as a Scaling Constant

$$C' = (1 + a_1 + a_2 + \dots + a_r)^{-1} \tag{2.38}$$

The scaling constant, C, ensures two necessary impulse sequence properties. First, it ensures that the output of the shaper does not violate actuator constraints. Each of the impulses in any shaping sequence are always less than one. Second, it ensures that the shaped steady state response will equal the steady state response of the un-shaped system, since the sum of all the impulses is equal to one.

2.3 Singhose Input Shaping Techniques

Singer [12] presented a vector diagram representation of input shaping and a method of finding the residual vibration of an input shaper from a vector diagram. Singhose [21, 22, 26] has extended the work of Singer et. al. on input shaping using the vector diagram representation. Singhose used vector diagrams to design input shapers that increase the level of insensitivity to modeling errors in frequency and damping, called Extra-Insensitive input shaping.

First Singer's method for representing input shapers using vector diagrams will be briefly discussed. The vectors are expressed using polar coordinates (r and θ). Where r represents the magnitude of an impulse and θ represents the damped or undamped natural frequency and impulse time. For the undamped case, a vector diagram can be defined by setting r_j equal to the impulse amplitude of the j_{th} impulse and the phase, θ_j $\omega\Delta T_j$. Where ΔT_j is the time delay from zero and ω is the undamped natural frequency of a system.

Figure 2.9 shows that the origin of each vector is at zero and that the first impulse, A_1, at time equal zero lies along the x-axis. Delays, θ_j's, are defined as a counter clockwise rotations from the positive x-axis. Residual vibration is defined by finding the resultant, R, from the sum of the two vectors (A_1, 0) and (A_2, θ) as shown in figure 2.10.

Figure 2.10 illustrates the concept of graphically determining residual vibration. A_r represents the amplitude of the vibration and θ_r represents the phase of the vibration.

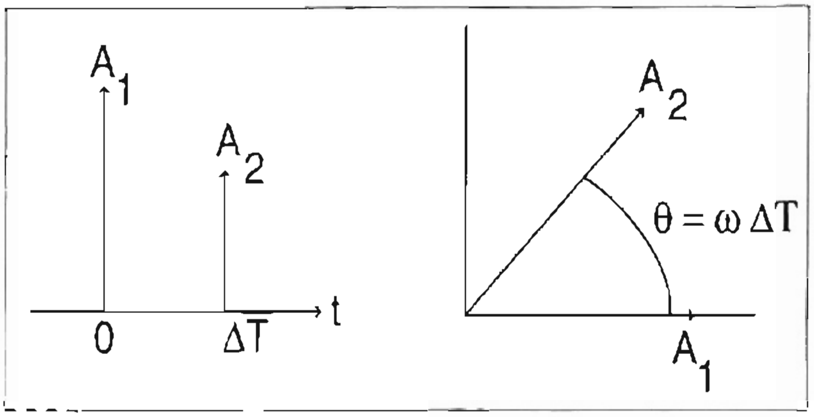

Figure 2.9: Vector Diagram Equivalent to a Two Impulse Input Shaper

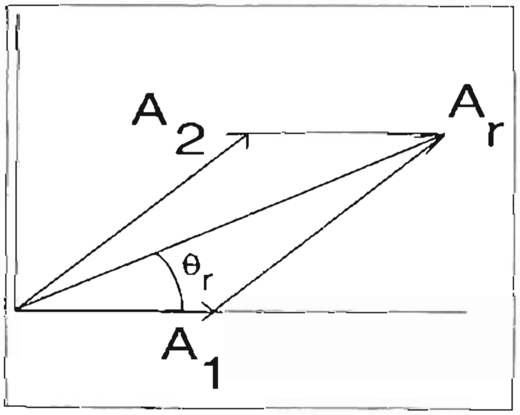

Figure 2.10: Formulation of Resultant Vector

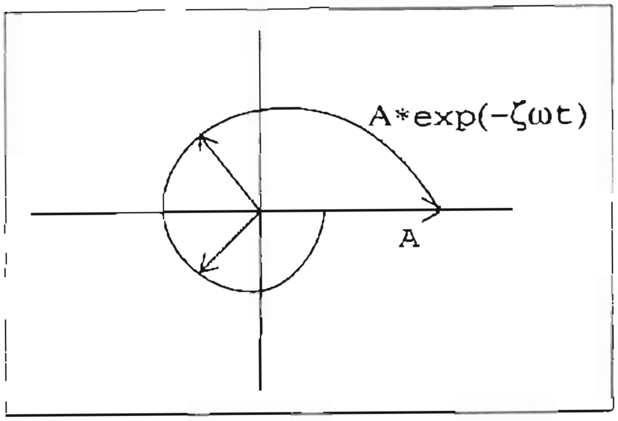

Figure 2.11: Decaying Amplitude of an Impulse Due to Damping

Damping effects the vector diagram representation in two ways. The undamped natural frequency can no longer be used to represent the phase, instead a system's damped natural frequency, $\omega_d = \omega\sqrt{1-\zeta^2}$, and impulse time(delay) represent the phase.

$$\theta = \omega\sqrt{1-\zeta^2}\Delta T \tag{2.39}$$

Also the damping causes impulses to decay exponentially with time. Consider the two impulse case shown in figure 2.5. The two impulses are 180° out of phase and the impulse A_2 has decayed with time with respect to A_1. This decay can be represented by equation 2.40 and by the spiral shown in figure 2.11.

$$A_2 = A_1 e^{-\zeta\omega t} \tag{2.40}$$

For either the undamped case or the damped case, vector diagrams can be used to graphically find the level of residual vibration. If no resultant exist after summing the vectors then no vibration occurs after the last impulse.

Modeling errors, primarily in frequency, cause the summing of the vectors to result in some level of residual vibration. Errors in damping are neglected here because Singer [12] showed that errors in frequency are more significant. Errors in frequency can be represented as a phase shift using vector diagrams. If ω_o is the nominal system frequency and ω is the design frequency then the phase shift, ϕ, is defined as:

$$\phi_j = (\omega - \omega_o)\Delta T_j \tag{2.41}$$

Some error in system parameters are most likely to occur in any real system. Therefore, shapers that are more insensitive to modeling errors must be derived.

2.3.1 An Extension of the Vector Diagram Approach

Using a vector diagram, Singhose [21, 22] showed that impulses that are not integer multiples of half a period of oscillation ($\theta = i\pi$) can widen the insensitivity curve at the 5% allowable residual vibration level. The drawback of this change is that the residual vibration curve is not symmetric about $\omega/\omega_o = 1$, when the design frequency perfectly matches the system frequency.

To widen the insensitivity symmetrically about $\omega/\omega_o = 1$, Singhose [21, 22, 26] defined a set of constraints that relaxed the zero residual vibration constraint defined by Singer [12]. Insensitivity is defined by the width of the sensitivity curve at 5% residual vibration($V = 0.05$). Therefore, the constraint on residual vibration will be relaxed to allow 5% residual vibration at $\omega/\omega_o = 1$.

The steps for finding a One Hump EI input shaper are as follows: 1.) Set the number of impulses to three (A_1, A_2, A_3). 2.) Define the phase shifts for A_1, A_2, and A_3 to be $\theta_j = 0$, π, and 2π, respectively(θ_3 must be twice θ_2 to satisfy the symmetry constraint). 3.) Solve for the impulse amplitudes such that there exist 5% residual vibration at $\omega/\omega_o = 1$ and that the residual vibration is zero above and below ω_o, ω_{hi} and ω_{lo} respectively. Figure 2.12 illustrates the structure for a One Hump EI input shaper.

Using these steps for the undamped case Singhose found the impulse sequence:

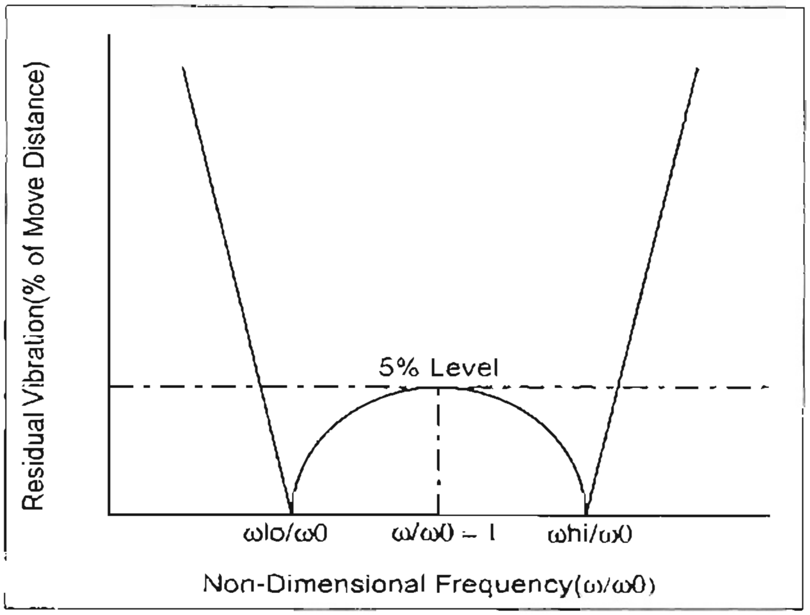
Residual Vibration(% of Move Distance)
5% Level
ωlo/ω0
ω/ω0 = 1
ωhi/ω0
Non-Dimensional Frequency(ω/ω0)

Figure 2.12: Diagram of Setup for One Hump EI Input Shaper

$$| A_1 |= \frac{1+V}{4} \quad @ \quad \theta_1 = 0 \tag{2.42}$$

$$| A_2 |= \frac{1-V}{2} \quad @ \quad \theta_2 = \pi \tag{2.43}$$

$$| A_3 |= \frac{1+V}{4} \quad @ \quad \theta_3 = 2\pi \tag{2.44}$$

where V is the level of desired residual vibration at $\omega/\omega_n = 1$. Notice that this definition sets the resultant equal to the level of residual vibration.

$$| A_1 | - | A_2 | + | A_3 | = V \tag{2.45}$$

Recall that $\theta_j = \omega \Delta T_j$, then the vectors in equations 2.42, 2.43, and 2.44 can be converted into an impulse sequence.

$$A_1 = \frac{1+V}{4} \quad @ \quad t_1 = 0 \tag{2.46}$$

$$A_2 = \frac{1-V}{2} \quad @ \quad t_2 = \frac{\pi}{\omega} \tag{2.47}$$

$$A_3 = \frac{1+V}{4} \quad @ \quad t_3 = \frac{2\pi}{\omega} \tag{2.48}$$

Closed form solutions for a single hump EI shaper cannot be determined for the case when damping is not neglected [22]. A numerical solution can be found for a range of damping ratios and levels of residual vibraion. Singhose calculated a solution for $0 \leqslant \zeta \leqslant 0.3$ and $0 \leqslant V \leqslant 0.15$. The solution he found is a function of ζ, V, and ω. The impulse amplitudes and times are shown in the following equations:

$$A_1 = 0.2497 + 0.2496V + 0.8001\zeta + 1.233V\zeta + 0.496\zeta^2 + 3.173V\zeta^2 \tag{2.49}$$

$$A_2 = 1 - (A_1 + A_3) \tag{2.50}$$

$$A_3 = 0.2515 + 0.2147V - 0.8325\zeta + 1.415V\zeta + 0.8518\zeta^2 - 4.901V\zeta^2 \tag{2.51}$$

$$T_1 = 0 \tag{2.52}$$

$$T_2 = (0.5 + 0.4616V\zeta + 4.262V\zeta^2 + 1.756V\zeta^3 - 8.578V^2\zeta - 108.6V^2\zeta^2 + 337V^2\zeta^3)T_d \tag{2.53}$$

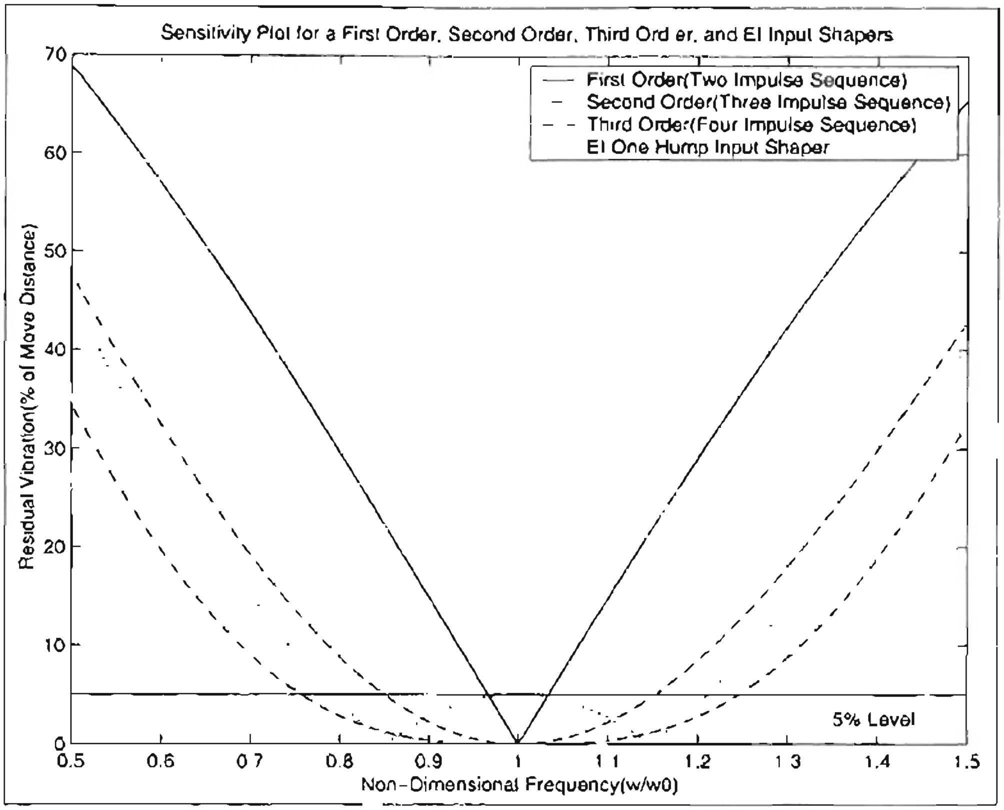

Figure 2.13: Sensitivity Comparison of Singer's First, Second, and Third Order Shapers to Singhose's One Hump EI

$$T_3 = T_d \tag{2.54}$$

Notice that the solution to a single hump EI shaper is the same length, T_d, as the ZVD or second order shaper. Where, T_d is the damped period of oscillation.

$$T_d = \frac{2\pi}{\omega\sqrt{1-\zeta^2}} \tag{2.55}$$

The primary advantage of a one hump EI input shaper is that it has the same duration as a second order(ZVD) input shaper but is more robust to modeling errors. Figure 2.13 shows a comparison of sensitivity curves for Singer's First(ZV), Second(ZVD), and Third(ZVDD) order shapers to Singhose's One Hump EI shaper.

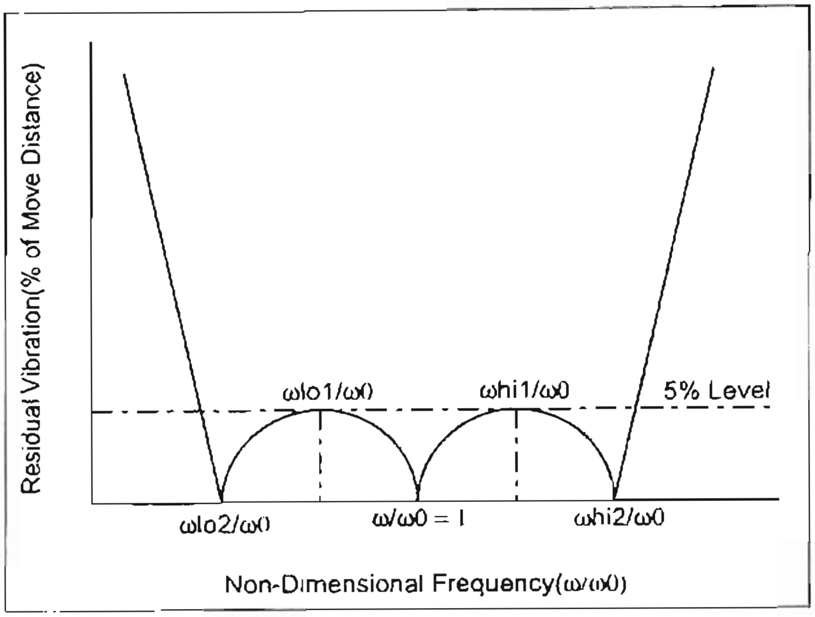
Residual Vibration(% of Move Distance)
ωlo1/ω0
ωhi1/ω0
5% Level
ωlo2/ω0
ω/ω0 = 1
ωhi2/ω0
Non-Dimensional Frequency(ω/ω0)

Figure 2.14: Diagram of Setup for Two Hump EI Input Shaper

Singhose also developed multiple hump shapers to widen the region of insensitivity more than the single hump. He developed two and three hump input shapers that allow a specified level of residual vibration at number of frequencies. the residual vibration is required to be zero at the system's natural frequency for a two hump shaper. It allows a certain level of residual vibration above, ω_{hi_1}, and below, ω_{lo_1}, the system natural frequency. Above ω_{hi_1} and below ω_{lo_1} the residual vibration is required to be zero at ω_{hi_2} and ω_{lo_2} as show in figure 2.14.

The three hump shaper can be described in a similar fashion, except it allows some vibration at the system frequency. Figure 2.15 illustrates the constraints of a Three Hump EI input shaper.

Figures 2.12. 2.14 and 2.15 show that the sensitivity curves are always symmetric about the system frequency. Therefore, an odd number of humps the shaper must allow some residual vibration at the system natural frequency and for an even number of humps the residual vibration must be zero at the system frequency. Singhose et. al. [26] provides a detailed derivation of these two techniques. Singhose [26] also derived a curve fit solution to the two and three hump EI input shapers for $0 \leqslant \zeta \leqslant 0.3$ and $0 \leqslant V \leqslant 0.2$. The curve

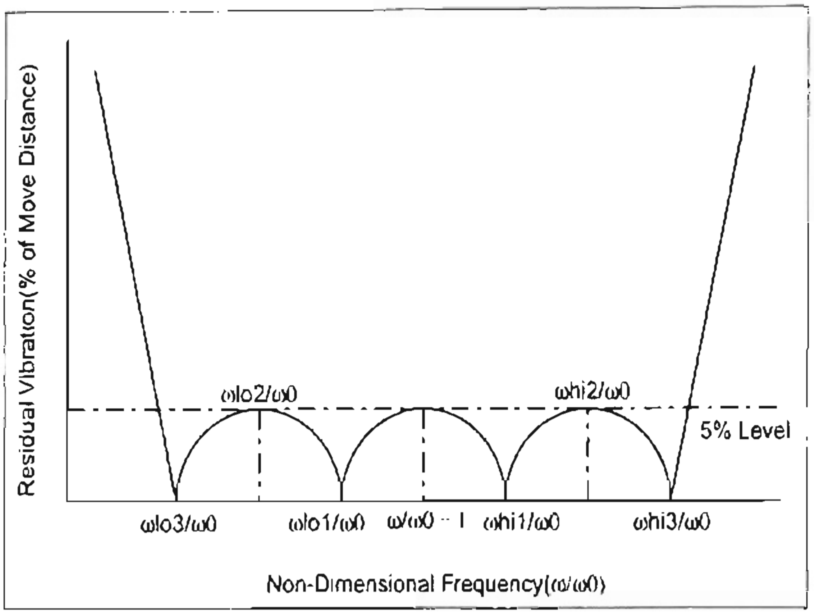
Residual Vibration(% of Move Distance)
ωlo2/ω0
ωhi2/ω0
5% Level
ωlo3/ω0
ωlo1/ω0
ω/ω0 = 1
ωhi1/ω0
ωhi3/ω0
Non-Dimensional Frequency(ω/ω0)

Figure 2.15: Diagram of Setup for Three Hump EI Input Shaper

fit is shown in table 2.3.

Figure 2.16 shows how a One and Two Hump EI input shaper compare to a Third order shaper designed using Singer's Technique. The One Hump shaper results in less robustness than a Third order shaper, but the Two Hump shaper has greater robustness than the Third order shaper. The Two Hump shaper and the Third order shaper have approximately the same time duration. Therefore, the Two Hump shaper allows system parameters to vary more with an equal amount of system rise time delay.

To compare the sensitivity of the One, Two, and Three Hump EI input shapers a plot of the three types designed for the same natural frequency and damping ratio is shown in figure 2.17.

As the number of humps increase the time duration and number of impulses also increase. EI input shaping is designed to maximize the level of robustness with each increase in time.

$$t_i = (M_0 + M_1\zeta + M_2\zeta^2 + M_3\zeta^3)T, T = 2\pi/\omega$$

$$A_i = M_0 + M_1\zeta + M_2\zeta^2 + M_3\zeta^2$$

Shaper		M_0	M_1	M_2	M_3
2 Hump	t_1	0	0	0	0
EI	t_2	0.49890	0.16270	-0.54262	6.16180
	t_3	0.99748	0.18382	-1.58270	8.17120
	t_4	1.49920	-0.09297	-0.28338	1.85710
	A_1	0.16054	0.76699	2.26560	-1.22750
	A_2	0.33911	0.45081	-2.58080	1.73650
	A_3	0.34089	-0.61533	-0.68765	0.42261
	A_4	0.15997	-0.60246	1.00280	-0.93145
3 Hump	t_1	0	0	0	0
EI	t_2	0.49974	0.23834	0.44559	12.4720
	t_3	0.99849	0.29808	-2.36460	23.3990
	t_4	1.49870	0.10306	-2.01390	17.0320
	t_5	1.99960	-0.28231	0.61536	5.40450
	A_1	0.11275	0.76632	3.29160	-1.44380
	A_2	0.23698	0.61164	-2.57850	4.85220
	A_3	0.30008	-0.19062	-2.14560	0.13744
	A_4	0.23775	-0.73297	0.46885	-2.08650
	A_5	0.11244	-0.45439	0.96382	-1.46000

Table 2.3: Curve Solutions to Two and Three Hump Extra Insensitive Input Shapers

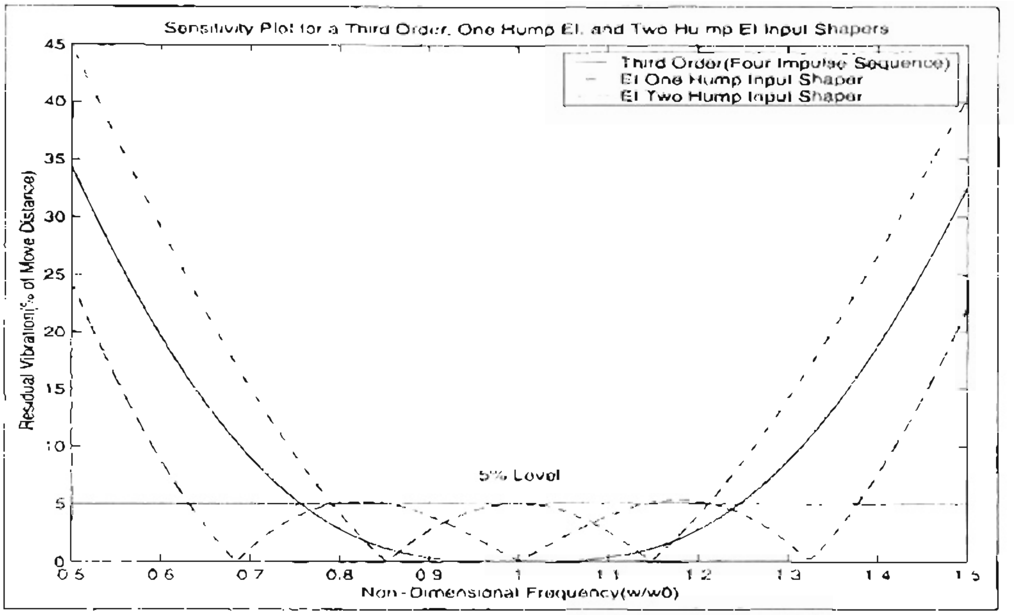

Figure 2.16: Sensitivity Comparison of Singer's Third Order Shaper to Singhose's One and Two Hump EI

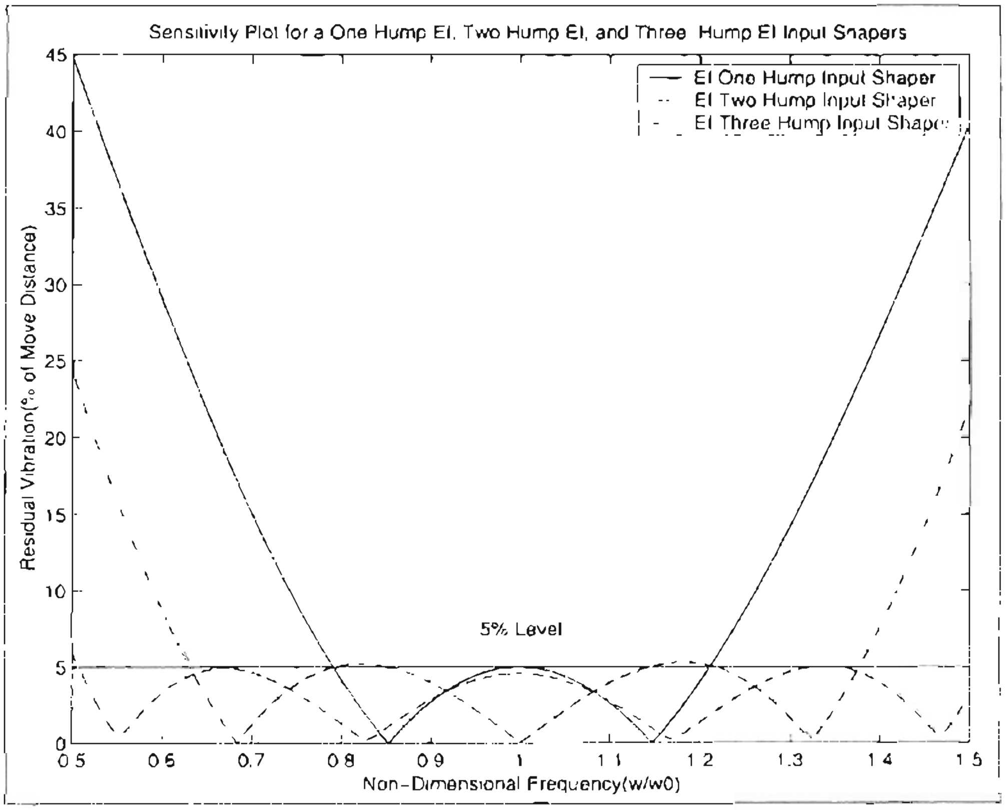

Figure 2.17: Sensitivity Comparison of Singhose's One, Two, and Three Hump EI

2.3.2 Singhose's Time-Optimal Negative Input Shaper

All the techniques up to this point require that all the impulse amplitudes have positive values, but if negative values are allowed the duration of an input shaper can be shortened. The material in this section was developed by W.E. Singhose, W.P. Seering, and N.C. Singer [30]. It is based on Singer's original constraint equations on the amplitude of residual vibration [12]. The constraints used in this technique can be categorized as constraints on the maximum allowable residual vibration, robustness to parameter uncertainty, time optimal solution constraints, and impulse amplitude constraints. Only zero-vibration(ZV) and zero-vibration-derivative(ZVD) shapers are developed for the negative input shapers developed in this section. Therefore, only the following two constraint equations on residual vibration are used:

$$V(\omega,\zeta) = \frac{\sqrt{(\sum_{j=1}^{n} A_j e^{\zeta\omega t_j}\cos(\omega t_j\sqrt{1-\zeta^2}))^2 + (\sum_{j=1}^{n} A_j e^{\zeta\omega t_j}\sin(\omega t_j\sqrt{1-\zeta^2}))^2}}{e^{\zeta\omega t_n}} \tag{2.56}$$

where V is ratio of vibration with input shaping to without input shaping. When V is required to be zero then the input shaper that meets that constraint is a zero-vibration input shaper or ZV shaper.

$$0 = \frac{d}{d\omega}\left(\frac{\sqrt{(\sum_{j=1}^{n} A_j e^{\zeta\omega t_j}\cos(\omega t_j\sqrt{1-\zeta^2}))^2 + (\sum_{j=1}^{n} A_j e^{\zeta\omega t_j}\sin(\omega t_j\sqrt{1-\zeta^2}))^2}}{e^{\zeta\omega t_n}}\right) \tag{2.57}$$

When an input shaper satisfies both equations 2.56 and 2.57 then it is said to be a zero-vibration-derivative input shaper or a ZVD input shaper. The variables in equations 2.56 and 2.57 are defined in section 2.1.1.

In order to minimize the duration of the input shaper a constraint on the time of the last impulse is implemented. It is defined as follows:

$$ShaperDuration = min(t_n) \tag{2.58}$$

where the n^{th} impulse is the last impulse of any sequence. Minimizing the duration of the input shaper minimizes the time lag that the input shaper induces into an input.

Time lag is not the only effect that convolving an arbitrary input with an input shaper. If the impulse amplitudes do not sum to one then the un-shaped input and the shaped input will not have the same set-point. Therefore, the following constraint is implemented:

$$\sum_{j=1}^{n} A_j = 1 \tag{2.59}$$

where this constraint applies to both input shapers with all positive impulses and input shapers with negative impulses. Satisfying the constraints in equations 2.56, 2.57, 2.58, and 2.59 can force the impulse amplitudes to positive and negative infinity. This was previously eliminated by requiring that all the impulse amplitudes are positive. For input shapers that allow negative and positive impulse sequences, two additional constraint are required and they are defined in section 2.3.3 and 2.3.4.

Another robustness constraint used to widen the range of insensitivity to errors in frequency is the Extra Insensitive (EI) constraint. Instead of requiring zero residual vibration at the system natural frequency some low level ($V = 5\%$) is allowed for the ZV constraint equation 2.56. To widen the sensitivity curve frequencies (ω_{lo} and ω_{hi}) on each side of the natural frequency are required to have zero residual vibration. The frequency ω_{lo} is an unknown frequency below the modeling frequency, ω_m and ω_{hi} is an unknown frequency above ω_m. Therefore, a set of two of the ZV constraints 2.56 are required. When the constraint equations are solved for ω_m the impulse times, amplitudes, and the unknown frequencies (ω_{lo} and ω_{hi}) are solved. This is the same shaper presented in section 2.3.1 except that the impulse amplitudes are defined using the constraints in sections 2.3.3 and 2.3.4.

2.3.3 Unity Magnitude Constraint (UM)

The first constraint is that the impulse amplitudes must be equal to one or minus one, also called unity magnitude. The sequence always starts with an impulse of one and the impulses switch between minus one and one after the first impulse. This constraint can be represented as:

$$A_j = (-1)^{j-1} \quad j = 1 \ldots n \tag{2.60}$$

This constraint guarantees that the shaped input will never exceed actuator limits, given that the un-shaped input does not exceed actuator limits. The last impulse of a unity magnitude input shaper is always one. Therefore, n must be an odd integer multiple of one.

2.3.4 Partial-Sum Constraint (PS)

To improve the rise times of the Unity Magnitude input shapers, a new constraint was developed that limits the sum of the impulses to a magnitude P.

$$\left|\sum_{j=1}^{k} A_j\right| = P \quad k = 1 \ldots n \tag{2.61}$$

For a zero-vibration (three impulse) input shaper, the solution with the constraint equation 2.61 is:

$$A_1 = P, \quad A_2 = -2P, \quad A_3 = P - 1 \tag{2.62}$$

Partial Sum shapers can cause momentary periods of actuator saturation, but the majority of the partial sum shaped command signals remain below $\pm P * MAX$. MAX is defined as the maximum allowable un-shaped command level. For example, if $P = 1$ and $MAX = 1$ then $P * MAX = 1$. The shaped signal for the step input shown in figure 2.18 does not exceed $\pm P * MAX$. When the same PS input shaper is applied to the bang-bang input shown in figure 2.19 there are periods of actuator saturation.

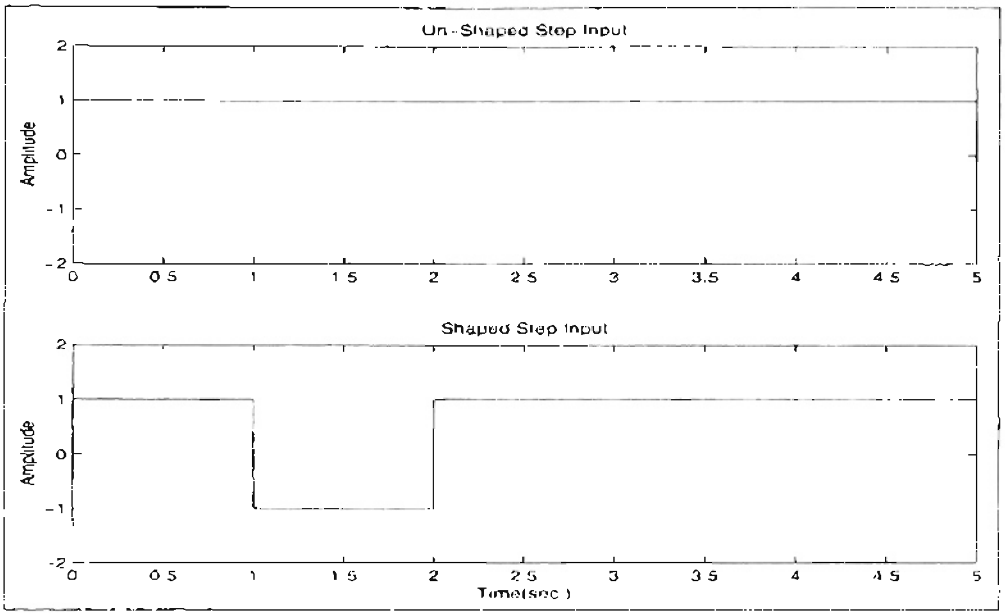

Figure 2.18: Step Input Shaped with a Partial Sum Input Shaper

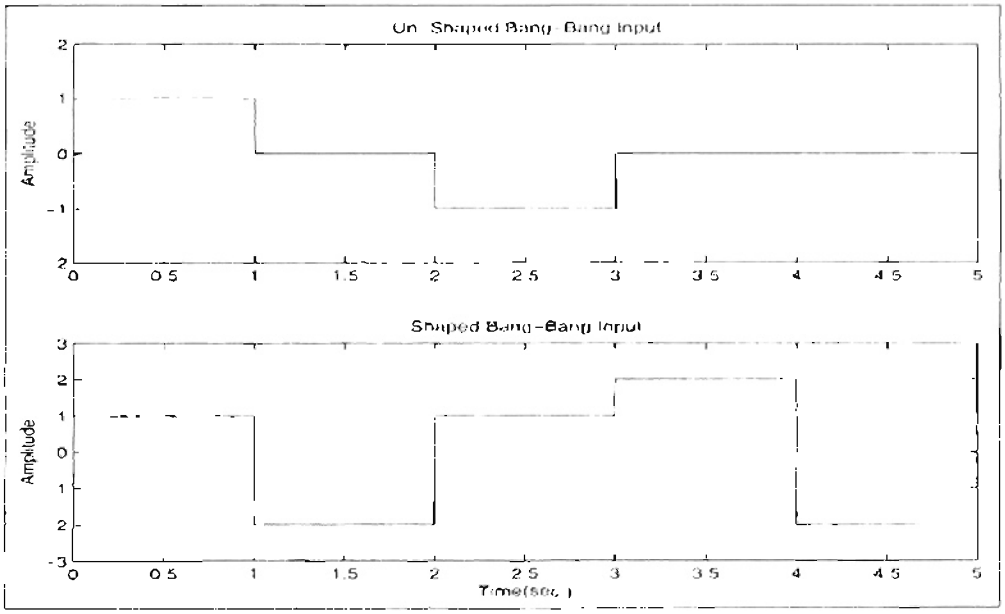

Figure 2.19: Bang Input Shaped with a Partial Sum Input Shaper

The duration of the actuator saturation a Partial Sum shaper causes depends on the required move distance, velocity limit, acceleration limit, natural frequency of the system, and the input shaper. Values of P larger than one are more likely to cause actuator saturation then when $P = 1$.

Singhose [30] developed a curve fit for the UM, PS, EI input shapers that is valid for $0 \leq \zeta \leq 0.3$ and is accurate to within 0.5% for the calculation of the impulse times. Table 2.4 and 2.5 list the curve fit solutions of the coefficients to several types of Partial Sum, Unity Magnitude, and Extra-Insensitive input shapers. The curve fits are for the impulse times only, unlike table 2.3.

$t_i = (M_0 + M_1\zeta + M_2\zeta^2 + M_3\zeta^3)T, T = 2\pi/\omega$						
Shaper	A_i	t_i	M_0	M_1	M_2	M_3
UM-ZV	1	t_1	0	0	0	0
	-1	t_2	0.16724	0.27242	0.20345	0
	1	t_3	0.33323	0.00533	0.17914	0.20125
PS-ZV	1	t_1	0	0	0	0
P=1	-2	t_2	0.20970	0.22441	0.08028	0.23124
	2	t_3	0.29013	0.09557	0.10346	0.24624
UM-ZVD	1	t_1	0	0	0	0
	-1	t_2	0.08945	0.28411	0.23013	0.16401
	1	t_3	0.36613	-0.08833	0.24048	0.17001
	-1	t_4	0.64277	0.29103	0.23262	0.43784
	1	t_5	0.73228	0.00992	0.49385	0.38633
PS-ZVD	1	t_1	0	0	0	0
P=1	-2	t_2	0.15234	0.23397	0.15168	0.21310
	2	t_3	0.27731	0.11147	0.04614	0.28786
	-2	t_4	0.63114	0.34930	0.11840	0.52558
	2	t_5	0.67878	0.19411	0.27432	0.48505

Table 2.4: Curve Solutions to ZV and ZVD Negative Input Shapers

$t_i = (M_0 + M_1\zeta + M_2\zeta^2 + M_3\zeta^3)T, T = 2\pi/\omega$

Shaper	A_i	t_i	M_0	M_1	M_2	M_3
UM-EI	1	t_1	0	0	0	0
V=5%	-1	t_2	0.09374	0.31903	0.13582	0.65274
	1	t_3	0.36798	-0.05894	0.13641	0.63266
	-1	t_4	0.64256	0.28595	0.26334	0.24999
	1	t_5	0.73664	0.00162	0.52749	0.19208
PS-EI	1	t_1	0	0	0	0
P=1	-2	t_2	0.15631	0.26556	0.05324	0.69457
V=5%	2	t_3	0.28080	0.13931	0.05627	0.75432
	-2	t_4	0.63427	0.34142	0.15371	0.32904
	2	t_5	0.68410	0.18498	0.31059	0.28565
2 Hump	1	t_1	0	0	0	0
UM-EI	-1	t_2	0.05970	0.31360	0.31759	1.5872
V=5%	1	t_3	0.40067	-0.08570	0.14685	1.6059
	-1	t_4	0.59292	0.38625	0.34296	1.2889
	1	t_5	0.78516	-0.08828	0.54174	1.3883
	-1	t_6	1.12640	0.20919	0.44217	0.30771
	1	t_7	1.18640	-0.02993	0.79859	0.10478
2 Hump	1	t_1	0	0	0	0
PS-EI	-2	t_2	0.12952	0.29981	0.08010	1.7913
P=1	2	t_3	0.27452	0.22452	-0.20059	1.8933
V=5%	-2	t_4	0.58235	0.51403	-0.00620	1.6106
	2	t_5	0.68355	0.26308	0.09029	1.7095
	-2	t_6	1.08870	0.39342	0.14197	0.48868
	2	t_7	1.12080	0.25926	0.35816	0.35035

Table 2.5: Curve Solutions to Extra Insensitive Negative Input Shapers

Chapter 3

Model Description

The dynamic model used in this research project is a torsional dynamic plant manufactured by Educational Control Products. The plant consists of three rotating disks connected by two flexible rods. These rods act as torsional springs and the drive disk is connected to the motor by a rigid belt drive system. Each disk is equipped with an optical encoder, but in simulations and experiments only the encoder on disk 3 is used.

3.1 Assumptions

The rods connecting the three disks are assumed to be linear springs. Though there are no dampers present in the system, light damping is assumed to be acting on each disk and the damping is also linear. All system properties such as inertia of the disks(static payload), spring rates, and damping coefficients are assumed to time invariant.

The effect of motor cogging is considered to make the system model more practical. Motor cogging is simplified to act as a pseudo linear spring. Equation 3.1 shows how motor cogging is approximated and simplified [10]. Figure 3.2 shows motor cogging acting as a spring and it is assumed to be time invariant.

$$T_{k1} \cong -\tilde{k}_1 sin(n\theta_1 + \phi) \cong -\tilde{k}_1 n\theta_1 \triangleq -k_1\theta_1 \tag{3.1}$$

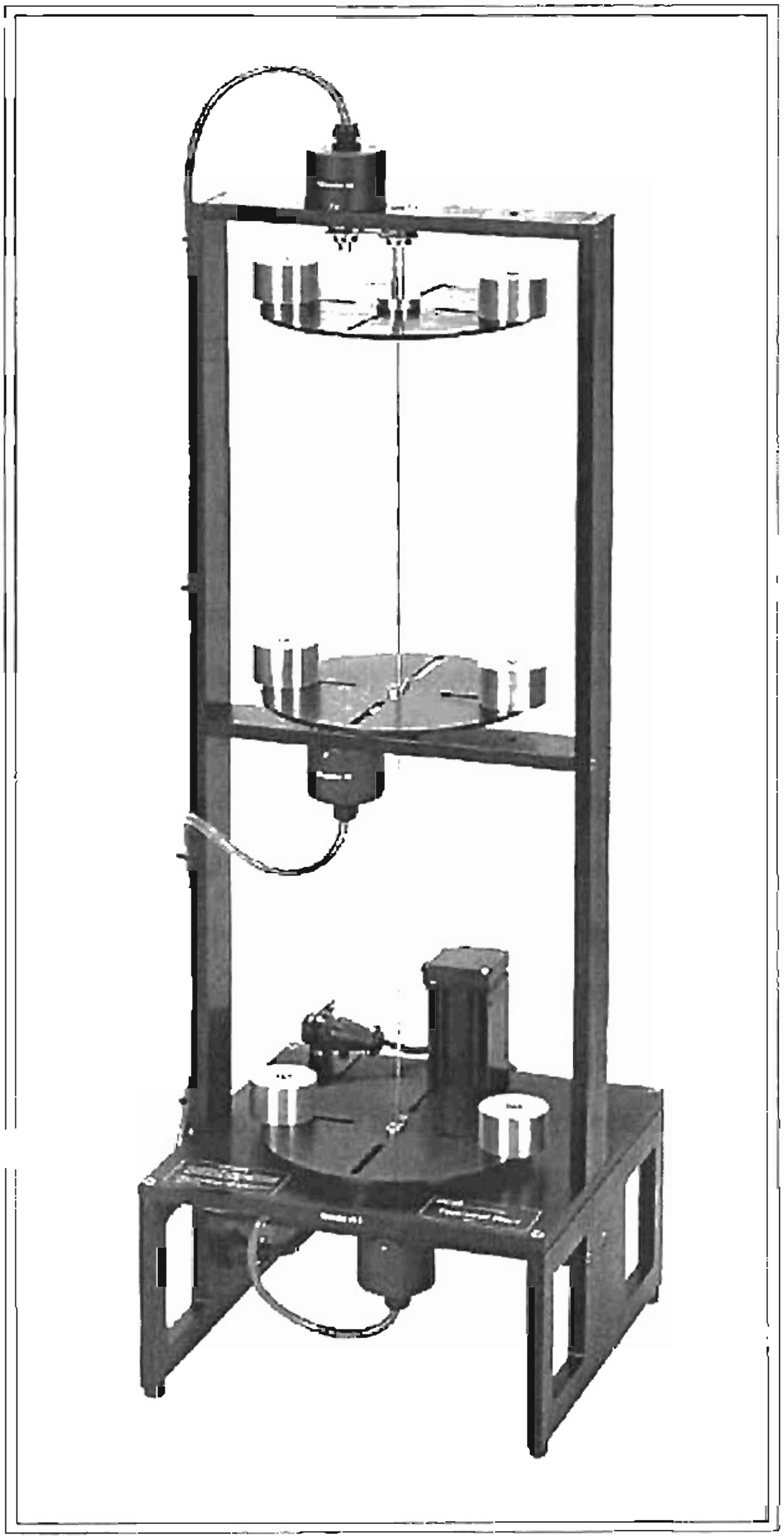

Figure 3.1: Torsional Apparatus

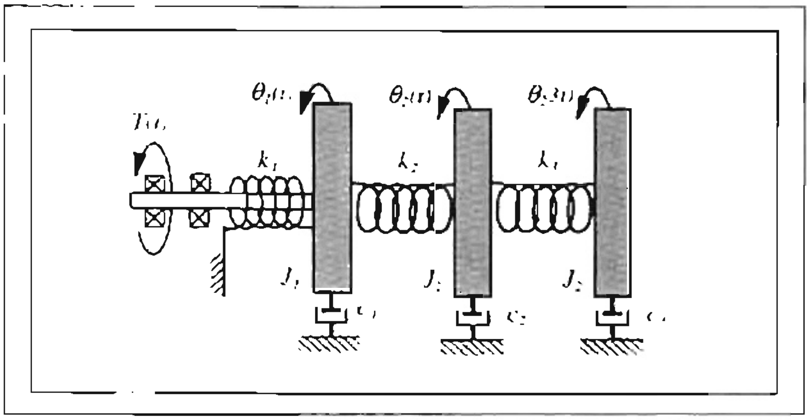

Figure 3.2: Model Diagram

3.2 Mathematical Model Derivation

The goal of this section is to find a linear time-invariant state space model of the system. A series of steps will be outlined to adequately describe the derivation of the model shown in figure 3.2.

The first step is to identify what elements are present in the system. The ECP torsional plant consists of three torsional springs, dampers, and inertias. Second, the energy storage elements are identified as the three springs(k_1, k_2, k_3) and three inertias(J_1, J_2, J_3). Third, free body diagrams are drawn for the energy storage elements to find the torques acting on those elements. The springs are drawn first because the goal is to find the equations of motion for the disks. The resulting torques acting on the springs are:

$$T_{k_1} = -k_1\theta_1 \tag{3.2}$$

$$T_{k_2} = k_2(\theta_1 - \theta_2) \tag{3.3}$$

$$T_{k_3} = k_3(\theta_2 - \theta_3) \tag{3.4}$$

where θ_1, θ_2, and θ_3 are the angular positions of each disk. Before finding the torques acting on the disk the energy dissipative elements(dampers) are found to be:

$$T_{c_1} = c_1\dot{\theta}_1 \tag{3.5}$$

$$T_{c_2} = c_2\dot{\theta}_2 \tag{3.6}$$

$$T_{c_3} = c_3\dot{\theta}_3 \tag{3.7}$$

where $\dot{\theta}_1$, $\dot{\theta}_2$, and $\dot{\theta}_3$ are the angular velocity of each disk. From the free body diagrams for each disk, the torques can be summed to find the equations of motion. The torque sums on each disk are determined to be:

$$J_1\ddot{\theta}_1 = T + T_{k_1} - T_{k_2} - T_{c_1} \tag{3.8}$$

where T is the input torque,

$$J_2\ddot{\theta}_2 = T_{k_2} - T_{c_2} - T_{k_3} \tag{3.9}$$

$$J_3\ddot{\theta}_3 = T_{k_3} - T_{c_3} \tag{3.10}$$

where $\ddot{\theta}_1$, $\ddot{\theta}_2$, and $\ddot{\theta}_3$ are the angular velocities of each disk. Substituting the torque equations found for the springs and dampers into the torque sums for the disks results in the following equations:

$$J_1\ddot{\theta}_1 = -(k_1 + k_2)\theta_1 - c_1\dot{\theta}_1 + k_2\theta_2 + T \tag{3.11}$$

$$J_2\ddot{\theta}_2 = k_2\theta_1 - (k_2 + k_3)\theta_2 - c_2\dot{\theta}_2 + k_3\theta_3 \tag{3.12}$$

$$J_3\ddot{\theta}_3 = k_3\theta_2 - k_3\theta_3 - c_3\dot{\theta}_3 \tag{3.13}$$

Fourth, a state vector is defined in order to find the equations of motion in state space form shown in equation 3.14. The state vector is defined in equation 3.15 as,

$$\dot{x} = Ax + Bu \tag{3.14}$$

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^T = \begin{bmatrix} \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 \end{bmatrix}^T \tag{3.15}$$

Take the derivative of the state vector,

$$\begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & \dot{x}_4 & \dot{x}_5 & \dot{x}_6 \end{bmatrix}^T = \begin{bmatrix} \dot{\theta}_1 & \ddot{\theta}_1 & \dot{\theta}_2 & \ddot{\theta}_2 & \dot{\theta}_3 & \ddot{\theta}_3 \end{bmatrix}^T \tag{3.16}$$

The equations of motion using the state vector and its derivative become,

$$\dot{x}_1 = x_2 \tag{3.17}$$

$$\dot{x}_2 = \frac{-(k_1 + k_2)}{J_1}x_1 - \frac{c_1}{J_1}x_2 + \frac{k_2}{J_1}x_3 + \frac{1}{J_1}T \tag{3.18}$$

$$\dot{x}_3 = x_4 \tag{3.19}$$

$$\dot{x}_4 = \frac{k_2}{J_2}x_1 - \frac{(k_2+k_3)}{J_2}x_4 - \frac{c_2}{J_2}x_4 + \frac{k_3}{J_2}x_5 \tag{3.20}$$

$$\dot{x}_5 = x_6 \tag{3.21}$$

$$\dot{x}_6 = \frac{k_3}{J_3}x_3 - \frac{k_3}{J_3}x_5 - \frac{c_3}{J_3}x_6 \tag{3.22}$$

Equations 3.17, 3.18, 3.19, 3.20, 3.21, and 3.22 can represented in state space form as,

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-(k1+k2)}{J1} & \frac{-c1}{J1} & \frac{k2}{J1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k2}{J2} & 0 & \frac{-(k2+k3)}{J2} & \frac{-c2}{J2} & \frac{k3}{J2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k3}{J3} & 0 & \frac{-k3}{J3} & \frac{-c3}{J3} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{J_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} T \tag{3.23}$$

The system parameters were approximated experimentally by ECP and the results of those experiments are shown in table 3.1. Using these approximations in simulation to find the open loop mode shapes confirms that they are good estimates because the experimental frequency response closely matches the simulation frequency response.

Using the parameters found by ECP, MATLAB codes were written to build the model and find the open loop modes. The code to build the model can be found in appendix A.1 and the code used to find the open loop mode shapes is in appendix A.2. The modes found from the model are flexible modes since the model is not a rigid body. After running model.m and mode.m the three modal frequencies and damping ratios are determined:

- $\zeta_1 = 0.0076$ & $\omega_1 = 7.1754 Rad/sec$

J_1	$0.01063\ kg - m^2$
J_2	$0.01063\ kg - m^2$
J_3	$0.01063\ kg - m^2$
c_1	$0.027\ N - m/(rad/s)$
c_2	$0.002\ N - m/(rad/s)$
c_3	$0.002\ N - m/(rad/s)$
k_1	$1.38\ N - m/rad$
k_2	$1.38\ N - m/rad$
k_3	$1.38\ N - m/rad$

Table 3.1: System Parameters

- $\zeta_2 = 0.0254$ & $\omega_2 = 20.1079 Rad/sec$

- $\zeta_3 = 0.0173$ & $\omega_3 = 28.9969 Rad/sec$

The modal frequencies were verified approximately by running a sine sweep over the range indicated from the simulation results. Sine sweeps can be performed on the ECP hardware using the Dynamics Executive described in section 3.4.

3.3 Hardware Description

The plant described in section 3.2 is manufactured by Educational Control Products (ECP). ECP provides the electromechanical plant, system interface software, DSP based controller/data acquisition board, and the input/output electronics (drive electronics). The software will be described in section 3.4 and the remainder of this section will be devoted to the hardware.

The base disk or disk one is the driven disk and it is driven by a brushless servo motor connected with a belt and pulley system with 3:1 gear reduction. The position of the first disk is measured by an axially mounted optical encoder. Positions of disk two and disk three are measured by radially mounted encoders. The encoders on disk two and three are mounted radially with a belt and pulley system with 1:1 gear reduction. All three encoders have a resolution of 16,000 encoder counts per revolution. The plant has the flexibility of allowing the user to select many different configurations [10]. Mathematical models are provided for many of the different configurations in the ECP manual [10]. The three degrees of freedom configuration will be used for all simulations and experiments.

The input/output electronics or drive electronics box contains the power supply, auxiliary digital-to-analog converter readouts, and the servo amplifier. The servo amplifier converts the voltage signal from the controller board into a current signal and sends the motor the current signal. The motor then transforms the current signal into a torque. A more detailed description of the power electronics is provided in the ECP manual [10].

The DSP board contains the encoder pulse decoders, D/A converters, and the realtime control algorithm downloaded from the software. The controller is defined in the software and then downloaded into the DSP board. The board automatically executes the control algorithm at a specified sampling rate and the board also executes any command signals that a user specifies. The minimum sampling period of the controller is 0.884 ms or a maximum servo loop closure computation rate of 1.131 kHz. For further information on the DSP board refer to the ECP manual [10]. Some of the hardware gains and ratios are listed in table 3.2. They may be used if a user wishes to model the mechanical model with the power electronics.

3.4 Software Packages

ECP provides three software packages to allow a user to interface with the DSP board: Executive, Dynamics Executive, and User Control Executive Program. The Executive program is designed to allow users to change the coefficients of the real time control algorithm in order to implement a desired control strategy. It also allows the use of many different

k_c	DAC Gain	$\frac{10V}{32768 DAC\ counts}$
k_a	Servo Amp Gain	$\approx 2amp/v$
k_t	Servo Motor Torque Constant	$\approx 0.1N - m/amp$
k_P	Drive Pulley Ratio	$3 : 1(Disk : Motor)$
k_e	Encoder Gain	$\frac{16000}{2\pi\ radians}$
k_s	Controller Gain	$\frac{32 controller\ counts}{encoder\ or\ reference\ input\ counts}$
k_{hw}	Hardware Gain	$k_{hw} = k_c k_a k_t k_P k_e k_s$

Table 3.2: Hardware Gains

inputs including a user defined input trajectory. The Dynamics Executive program is designed to study the dynamic characteristics of the system and to identify system parameters or mode shapes. The ECP manual [10] describes a number of different experiments for parameter identification for the numerous system configurations. The User Control program is designed for implementing a custom control algorithm in a "C-Like" code. ECP provides an example of how to code a state feedback controller in the User Control executive program. All software packages can plot real time data or data off-line after an input command has been executed. There are 24 acquired and derived variables available for plotting, and any real time data can be exported in a format that can be plotted in MATLAB.

The general format for implementing a control algorithm and executing a trajectory are simple. First, the system configuration must be selected by adding or adjusting the position of the masses. Also, a disk may be clamped to reduce the order of the system. Second, the system should be modeled mathematically to design a control algorithm. Third, after the control algorithm is designed it should be tested in simulation before implementing it on the hardware. Poor controller design can cause damage to the hardware and possible injury to the user. Finally, after the simulation tests the controller can be tested experimentally to compare results. The user must adjust the coefficients shown in figure 3.3 to a desired controller and them implement the algorithm. Then the software downloads the controller into the DSP board and the system is ready to receive a command input.

3.5 Derivation of Discrete Time Model

The linear time invariant/continuous time model has been derived in state space form. Since the goal is to reduce the level of residual vibration in a computer controlled system, the continuous time model should be discretized with the controller and with a sampling period that is representative of the computer hardware speed. For simulations and experiments a PD controller will be used to control the position of the third disk. PD control is used because of its effectiveness and functional simplicity. The gains for the PD controller are found by iteration on a trial-and-error basis. With the PD gains and the sampling rate specified, the discrete time model is found. A series of steps will be described to find the discrete time model.

First, the continuous time model is discretized with respect to the sampling period, T_s, using the command *c2d(sysc,*T_w*)* command in MATLAB, where *sysc* is the continuous time system matrix found using *ss(A,B,C,DD)*. Second, the forward path of the closed loop system is formed in SIMULINK with the discrete PD controller and the discrete time state space system found in the previous step. The SIMULINK diagram is shown in figure 3.4.

To find the overall system representation of figure 3.4 the *dlinmod(OLSSPDModel,*T_s*)* command finds the discrete system matrix for a specified SIMULINK model(OLSSPDModel.mdl) and sampling period(T_s). Third, the closed loop system matrix is found using *feedback(sysdt,1)*, where *sysdt* is the system matrix found in the previous step. Since *feedback* assumes negative feedback, 1 is the second command input. Finally, the closed loop damping and undamped natural frequencies of the system poles are found using *damp(sysdtcl)*, where *sysdtcl* is the closed loop discrete time system matrix.

These steps are executed at the beginning of each input shaper design script by *Discrete.m* shown in appendix A.3. The controller gains and sampling period are specified in

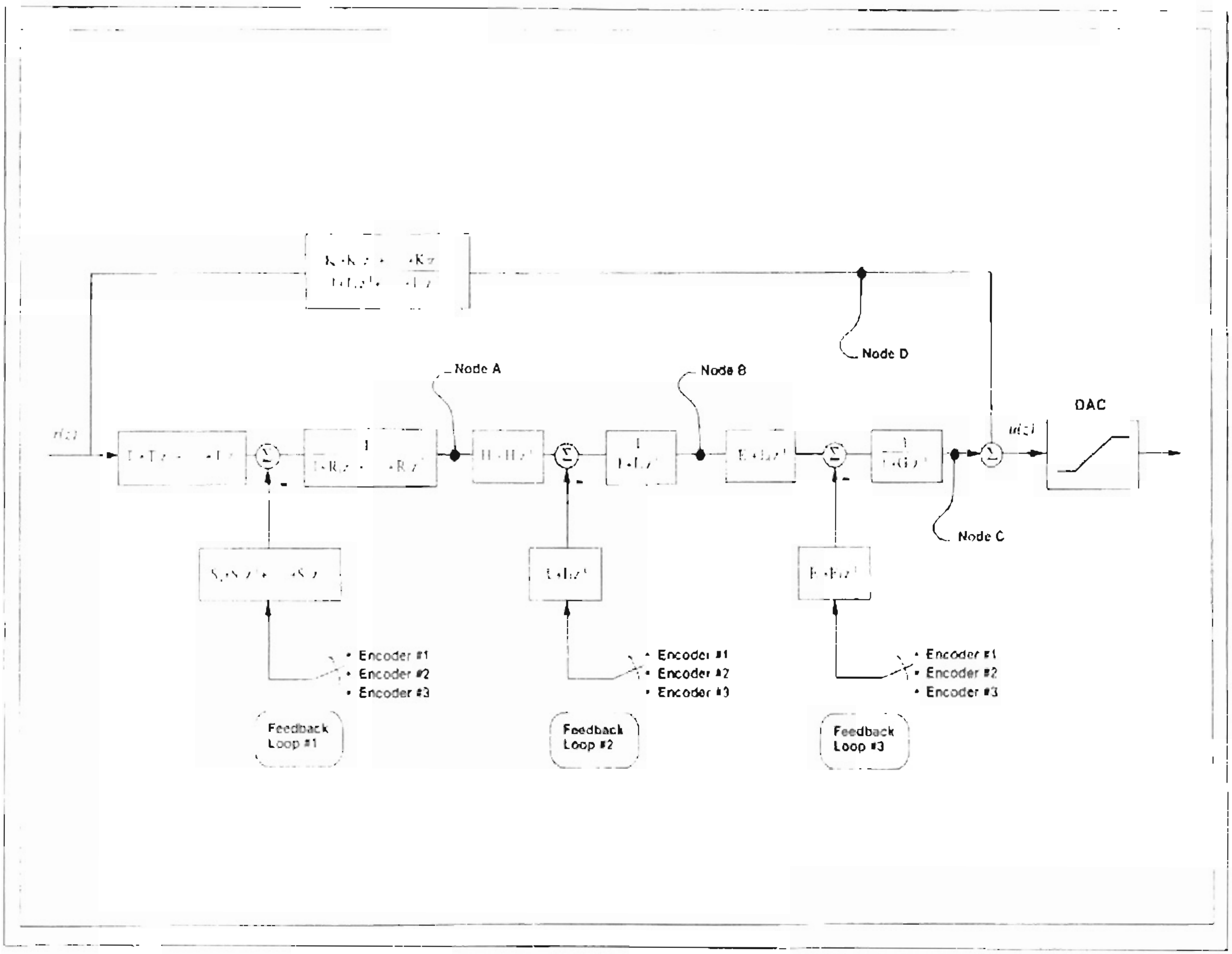

Figure 3.3: Real Time Control Algorithm

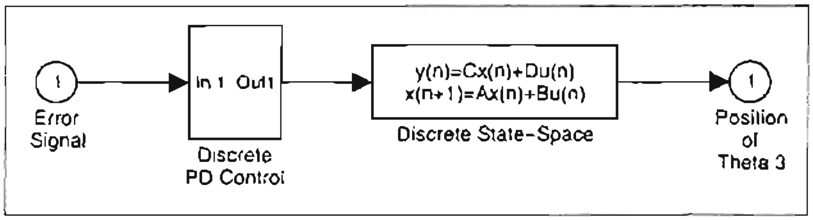

Figure 3.4: Discrete Time Forward Path

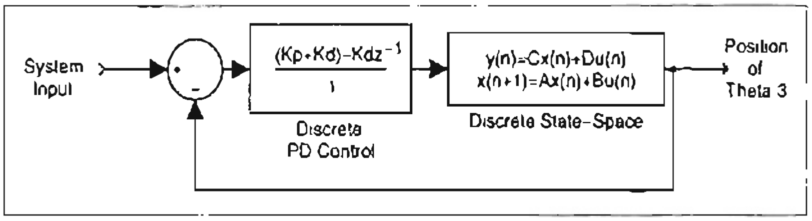

Figure 3.5: Discrete Time Closed Loop System

each input shaper design script before *Discrete.m* is executed.

The shaped commands generated by each input shaping technique are tested, in simulation, using the closed loop system before being evaluated experimentally on the ECP hardware. The system representation for simulation is shown in figure 3.5.

Chapter 4

Time Domain Constraint Equation Technique

This chapter covers the derivation of several input shapers using Dr. Singer's technique outlined in section 2.1, and an extension of his work modified to improve its ease of implementation in simulation and experiments. To illustrate the effect of each mode on the system response, impulse sequences are designed for each mode separately. Also, direct solutions for multiple mode systems are not considered because they have been found to be less robust than convolved solutions [29, 31]. The simulation results for the modified technique will be compared to the experimental results. Experimental results are obtained by executing the input command signal from simulation on the ECP hardware(Torsional Plant).

4.1 Derivation of a First Order(ZV) Input Shaper

The input used in all simulations is a unit step. This simplifies the convolution process required to shape the input. The first shapers derived are first order(two-impulse) shapers for each mode separately. This was done by using the equations defined in section 2.1.

First Order(ZV) Input Shaper					
Mode	i	A_{i1}	A_{i2}	ΔT_{i1}	ΔT_{i2}
First	1	0.5244	0.4756	0 sec	0.4323 sec
Second	2	0.5284	0.4716	0 sec	0.1568 sec
Third	3	0.5137	0.4863	0 sec	0.1083 sec

Table 4.1: Impulse Amplitudes and Times For First Order Shaper

$$A_{i1} = \frac{1}{1 + K_i} \tag{4.1}$$

$$A_{i2} = \frac{K_i}{1 + K_i} \tag{4.2}$$

where,

$$K_i = e^{-\frac{\zeta_i \pi}{\sqrt{1-\zeta_i^2}}} \tag{4.3}$$

$$\Delta T_i = \frac{\pi}{\omega_i \sqrt{1 - \zeta_i^2}} \tag{4.4}$$

Now the impulses are defined for the i^{th} mode, so that later they may be convolved without confusion. For this three mode case i varies from 1 to 3, where the closed loop damping ratios and natural frequencies are: $\zeta_1 = 0.0310$, $\omega_1 = 7.2707 Rad/sec$, $\zeta_2 = 0.0362$, $\omega_2 = 20.0457 Rad/sec$, $\zeta_3 = 0.0175$, and $\omega_3 = 29.0160 Rad/sec$ for PD control with $K_p = 0.06$ and $K_d = 0.75$.

The impulse amplitudes and times are found by plugging the system parameters(ζ_i & ω_i) for each mode into equations 4.3 and 4.4. The results for the torsional plant are found in table 4.1 for each mode separately. Figure 4.1 shows a step input convolved with the three shapers given in table 4.1. It shows that input shapers for high frequency modes are

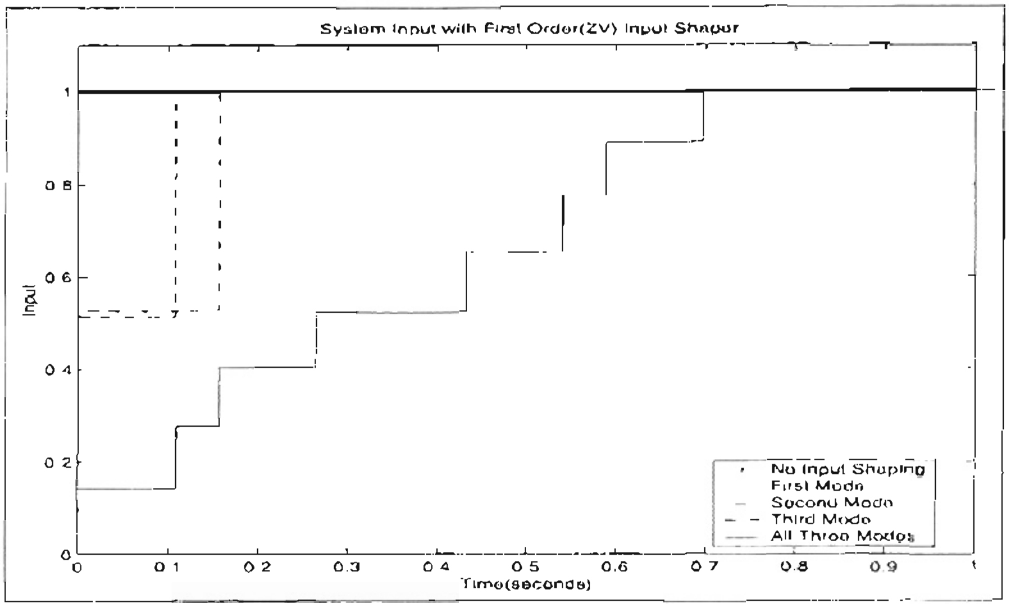

Figure 4.1: First Order Shaper Inputs

shorter in duration than input shapers for low frequency modes. This is a result of the time duration being half the period of oscillation of a particular mode. It also shows a step input convolved with an input shaper designed for all three modes, the all mode shaper will be derived later in this section.

Figure 4.2 shows the torsional system response to un-shaped and shaped inputs. It is easy to see that the low frequency mode dominates the response, but the input shaper designed for the low frequency mode alone results in substantial residual vibration.

To find a first order shaper for all three modes a series of steps will be outlined to show a method for convolving the shapers. First the equations for the inputs are found in continuous time. They are defined for the i^{th} mode as:

$$u_i(t) = A_{i1}\delta(t) + A_{i2}\delta(t - \Delta T_i) \tag{4.5}$$

Second, the Laplace transform of equation 4.5 is found to avoid having to convolve three continuous time equations since multiplication in the frequency domain is equivalent to convolution in continuous time. The Laplace domain representation of equation 4.5 is

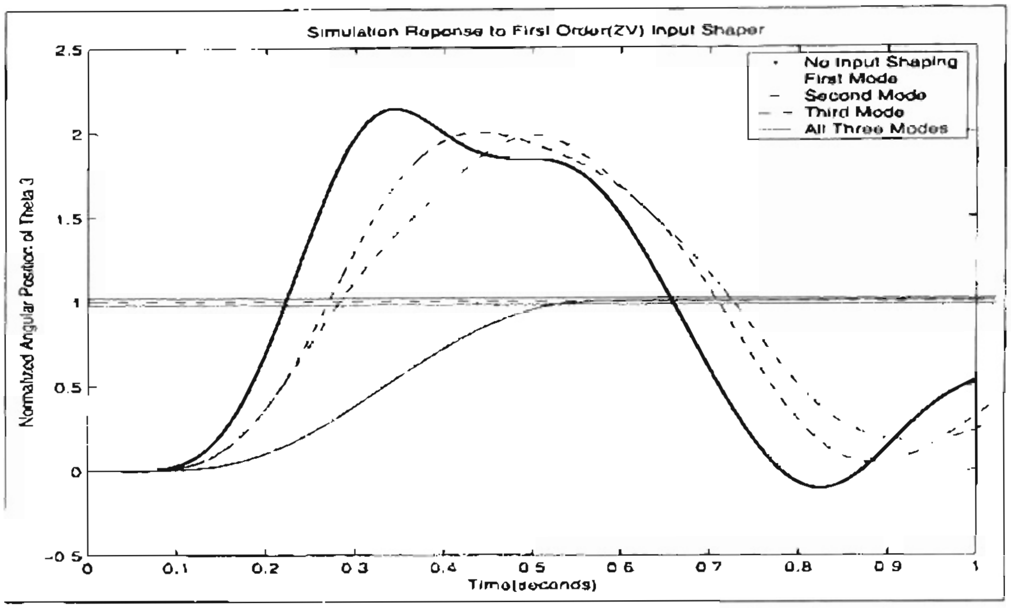

Figure 4.2: First Order Shaper Step Response

defined as:

$$U_i(s) = A_{i1} + A_{i2}e^{-\Delta T_i s} \tag{4.6}$$

Therefore, the equation for the multiple mode shaper is found by polynomial expansion of equation 4.6 for $U_1(s)U_2(s)U_3(s)$. Third, the result from multiplying the shapers in the Laplace domain is transformed back to continuous time by taking the inverse Laplace transform of $U_1(s)U_2(s)U_3(s)$. Finally the shaper can be defined in continuous time as:

$$u_1 u_2 u_3 = \begin{bmatrix} A_{11}A_{21}A_{31}\delta(t) + A_{12}A_{21}A_{31}\delta(t - \Delta T_1) \\ +A_{11}A_{22}A_{31}\delta(t - \Delta T_2) + A_{11}A_{21}A_{32}\delta(t - \Delta T_3) \\ +A_{12}A_{22}A_{31}\delta(t - \Delta T_1 - \Delta T_2) \\ +A_{12}A_{21}A_{32}\delta(t - \Delta T_1 - \Delta T_3) \\ -A_{11}A_{22}A_{32}\delta(t - \Delta T_2 - \Delta T_3) \\ +A_{12}A_{22}A_{32}\delta(t - \Delta T_1 - \Delta T_2 - \Delta T_3) \end{bmatrix} \tag{4.7}$$

Equation 4.7 shows that there are eight impulses resulting from the polynomial expansion of three first order polynomials or 2^n impulses where n is the number of modes present in the system. It also shows that the impulse times are not equally spaced for the convolved solution, which is inconvenient if it is to be implemented as a digital filter. A solution to this problem is presented in sections 4.3 and 4.4. Figure 4.1 shows the impulse amplitudes and times for equation 4.7 and figure 4.2 shows the response to the shaper designed for all three modes. Since the system parameters for simulation are known exactly, the shaper designed for all three modes completely eliminates all residual vibration. The parameters have been verified experimentally, but it is unlikely that they do not vary from the simulation model. Therefore, a second order shaper will be designed for all three modes to ensure a more robust output.

4.2 Derivation of a Second Order(ZVD) Input Shaper

Before finding a shaper that compensates for all three modes, the individual shapers are derived. The impulse amplitudes for a second order input shaper are defined for the i^{th} mode as shown for the first order case in section 4.1. For the three impulse case the impulse amplitudes and times are defined as:

$$A_{i1} = \frac{1}{1+2K_i+K_i^2} \quad @ \quad \Delta T_{i1} = 0 \tag{4.8}$$

$$A_{i2} = \frac{2K_i}{1+2K_i+K_i^2} \quad @ \quad \Delta T_{i2} = \frac{\pi}{\omega_i\sqrt{1-\zeta_i^2}} \tag{4.9}$$

$$A_{i3} = \frac{K_i^2}{1+2K_i+K_i^2} \quad @ \quad \Delta T_{i3} = 2\Delta T_{i2} \tag{4.10}$$

Second Order(ZVD) Input Shaper							
Mode	i	A_{i1}	A_{i2}	A_{i3}	ΔT_{i1}	ΔT_{i2}	ΔT_{i3}
First	1	0.2749	0.4988	0.2262	0 sec	0.4323 sec	0.8646 sec
Second	2	0.2792	0.4984	0.2224	0 sec	0.1568 sec	0.3136 sec
Third	3	0.2639	0.4996	0.2364	0 sec	0.1083 sec	0.2166 sec

Table 4.2: Impulse Amplitudes and Times For Second Order Shaper

where K_i is still defined by equation 4.3. Again, the system parameters are used to calculate the impulse amplitudes and times. The result for the closed loop plant is shown in table 4.2. With the individual input shapers derived the convolved solution can be determined. The procedure to find the convolved solution will be briefly covered again.

The continuous time representation for a second order input shaper contains three impulses and can be expressed as:

$$u_i(t) = A_{i1}\delta(t) + A_{i2}\delta(t - \Delta T_{i2}) + A_{i3}\delta(t - 2\Delta T_{i2}) \qquad (4.11)$$

The Laplace transform of equation 4.11 is:

$$U_i(s) = A_{i1} + A_{i2}e^{-\Delta T_{i2}s} + A_{i3}e^{-2\Delta T_{i2}s} \qquad (4.12)$$

The polynomial expansion of three second order input shapers results in a impulse sequence with 27 impulses and a duration equal to the sum of the three individual shapers($2\Delta T_{12} + 2\Delta T_{22} + 2\Delta T_{32}$). The result is defined by equation 4.13. For the three impulse case the number of impulses for the convolved solution is 3^n, where n is the number of modes.

$$u_1 u_2 u_3 = \left[\begin{array}{c}
A_{11}A_{21}A_{31}\delta(t) \\
+A_{12}A_{21}A_{31}\delta(t-\Delta T_{12}) + A_{12}A_{21}A_{31}\delta(t-2\Delta T_{12}) \\
+A_{11}A_{22}A_{31}\delta(t-\Delta T_{22}) + A_{11}A_{23}A_{31}\delta(t-2\Delta T_{22}) \\
+A_{11}A_{21}A_{32}\delta(t-\Delta T_{32}) + A_{11}A_{21}A_{33}\delta(t-2\Delta T_{32}) \\
+A_{11}A_{22}A_{33}\delta(t-\Delta T_{22}-2\Delta T_{32}) \\
+A_{11}A_{22}A_{32}\delta(t-\Delta T_{22}-\Delta T_{32}) \\
+A_{13}A_{23}A_{31}\delta(t-2\Delta T_{12}-2\Delta T_{22}) \\
+A_{13}A_{22}A_{31}\delta(t-2\Delta T_{12}-\Delta T_{22}) \\
+A_{13}A_{21}A_{33}\delta(t-2\Delta T_{12}-2\Delta T_{32}) \\
+A_{13}A_{21}A_{32}\delta(t-2\Delta T_{12}-\Delta T_{32}) \\
+A_{12}A_{23}A_{31}\delta(t-\Delta T_{12}-2\Delta T_{22}) \\
+A_{12}A_{22}A_{31}\delta(t-\Delta T_{12}-\Delta T_{22}) \\
+A_{12}A_{21}A_{33}\delta(t-\Delta T_{12}-2\Delta T_{32}) \\
+A_{11}A_{23}A_{32}\delta(t-2\Delta T_{22}-\Delta T_{32}) \\
+A_{12}A_{21}A_{32}\delta(t-\Delta T_{12}-\Delta T_{32}) \\
+A_{11}A_{23}A_{33}\delta(t-2\Delta T_{22}-2\Delta T_{32}) \\
+A_{12}A_{23}A_{32}\delta(t-\Delta T_{12}-2\Delta T_{22}-\Delta T_{32}) \\
+A_{12}A_{22}A_{33}\delta(t-\Delta T_{12}-\Delta T_{22}-2\Delta T_{32}) \\
+A_{12}A_{22}A_{32}\delta(t-\Delta T_{12}-\Delta T_{22}-\Delta T_{32}) \\
+A_{13}A_{22}A_{33}\delta(t-2\Delta T_{12}-\Delta T_{22}-3\Delta T_{32}) \\
+A_{13}A_{22}A_{32}\delta(t-2\Delta T_{12}-\Delta T_{22}-\Delta T_{32}) \\
+A_{12}A_{23}A_{33}\delta(t-\Delta T_{12}-2\Delta T_{22}-2\Delta T_{32}) \\
+A_{13}A_{23}A_{32}\delta(t-2\Delta T_{12}-2\Delta T_{22}-\Delta T_{32}) \\
+A_{13}A_{23}A_{33}\delta(t-2\Delta T_{12}-2\Delta T_{22}-2\Delta T_{32})
\end{array}\right] \qquad (4.13)$$

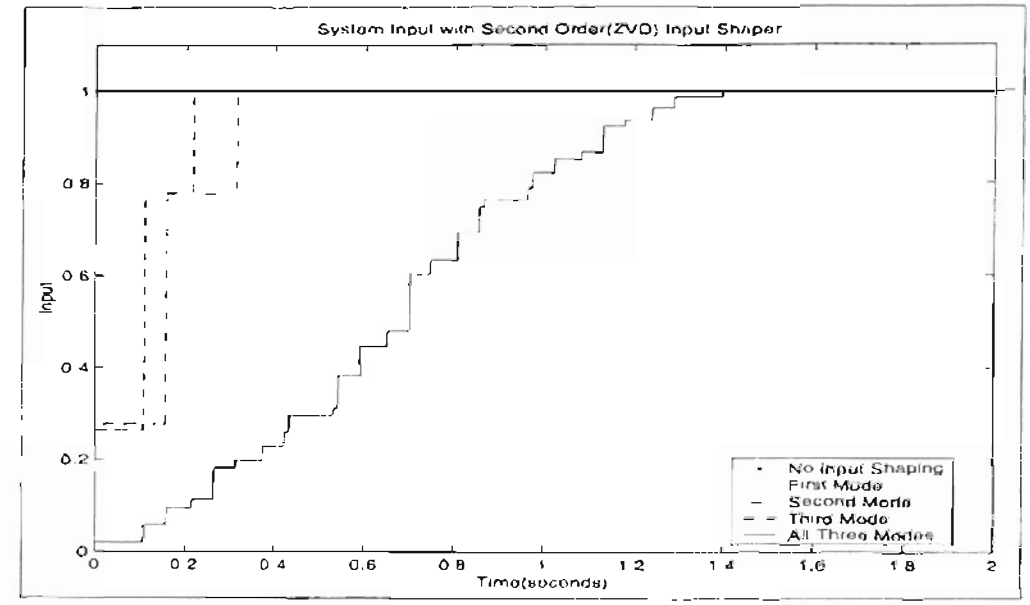

Figure 4.3: Second Order Shaper Inputs

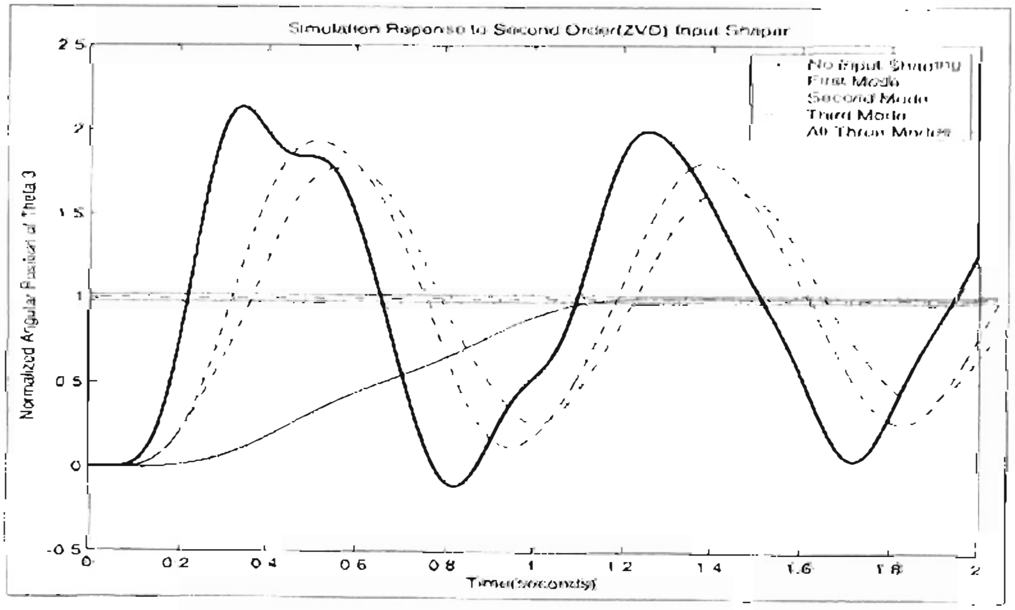

Figure 4.4: Second Order Shaper Step Response

4.3 Digital Two Mode First Order Shaper

A modified discrete time method of first order(ZV) input shaping is derived in this section. The first order(ZV) and second order(ZVD) input shapers could not be verified experimentally, and can not be implemented as a digital filter for the multiple mode case. Those limitations motivate the need for a discrete time method. Two cases are considered, second mode modified and third mode modified. The input shaper for the first mode remains unchanged for both cases since it has been shown to be the dominant mode. The procedure for both cases is identical and will be presented simultaneously.

The idea of this modified technique is to change the parameters($\zeta_{2,3}$ & $\omega_{2,3}$) of the second or third mode such that the impulse time of the first mode is an integer multiple of the impulse time of the second or third mode. For the torsional system, only two modes will be included in the design of this technique either the first and second(1/2) or first and third(1/3). Using the impulse times shown in table 4.1, the integer multiples for 1/2 case and 1/3 case are found to be 3 and 4, respectfully. The integer multiples are chosen such that the change in the impulse times(ΔT_{22} and ΔT_{32}) are minimized. The modified times for each case are shown below:

$$m\Delta T_{22} = \Delta T_{12}/3 = 0.1441 \tag{4.14}$$

$$m\Delta T_{32} = \Delta T_{12}/4 = 0.108075 \tag{4.15}$$

The procedure for the case when the second mode is modified is presented. Using the modified time for the second impulse, $m\Delta T_{22}$, the natural frequency and damping ratio of the second mode are modified with equation 4.16. The frequency is changed such that $\left(\frac{\pi}{m\omega_2 * m\Delta T_{22}}\right)^2$ is less than one. Then the modified damping ratio is calculated using equation 4.16.

$$m\zeta_2 = \sqrt{1 - \left(\frac{\pi}{m\omega_2 \times m\Delta T_{22}}\right)^2} \tag{4.16}$$

The integer multiple is determined by hand but the modified frequency and damping ratio is found using the following portion of a MATLAB script. This works for the case when the modified time is less than the original time, requiring the frequency to be increased. Also, the script minimizes the variation in the natural frequency but ignores the variation in damping ratio, since errors in frequency have been found to be more significant than errors in damping.

```
mT_22=T_12/3;
mw2=Wn(3,1);
counter=0.0001;
temp1=(pi/(mw2*mT_22))^2;
while temp1 > 1

    temp1=(pi/((mw2+counter)*mT_22))^2;

    counter=counter+0.0001;

end
mw2=mw2+counter;
mzeta2=sqrt(1-(pi/(mw2*mT_22))^2);
```

With the modified parameters known the impulse amplitudes can be calculated for the second mode and the equations for both input shapers can be found. Let the sampling period of the digital filter be,

$$T = m\Delta T_{22} \tag{4.17}$$

$$\therefore 3T = \Delta T_{12} \tag{4.18}$$

Now the input shaper equations for modes 1 and 2 can be found using the same sampling period. The input shaper for each mode is first expressed in the time domain. Then the sampling period is substituted for the time delays. The laplace transform of the time domain representation is found and then mapped the discrete frequency domain using $z = e^{Ts}$.

$$u_1(t) = A_{11}\delta(t) + A_{12}\delta(t - \Delta T_{12}) = A_{11}\delta(t) + A_{12}\delta(t - 3T) \tag{4.19}$$

$$U_1(s) = A_{11} + A_{12}e^{-3Ts} \tag{4.20}$$

$$U_1(z) = A_{11} + A_{12}z^{-3} = \frac{A_{11}z^3 + A_{12}}{z^3} \tag{4.21}$$

The modified first order input shaper for the second mode:

$$mu_2(t) = mA_{21}\delta(t) + mA_{22}\delta(t - m\Delta T_{22}) = mA_{21}\delta(t) + mA_{22}\delta(t - T) \tag{4.22}$$

$$mU_2(s) = mA_{21} + mA_{22}e^{-Ts} \tag{4.23}$$

$$mU_2(z) = mA_{21} + mA_{22}z^{-1} = \frac{mA_{21}z + mA_{22}}{z} \tag{4.24}$$

Equation 4.21 and 4.24 are used in SIMULINK as cascaded digital filters with the same sampling period. Multiplying the result of the equations is not necessary because

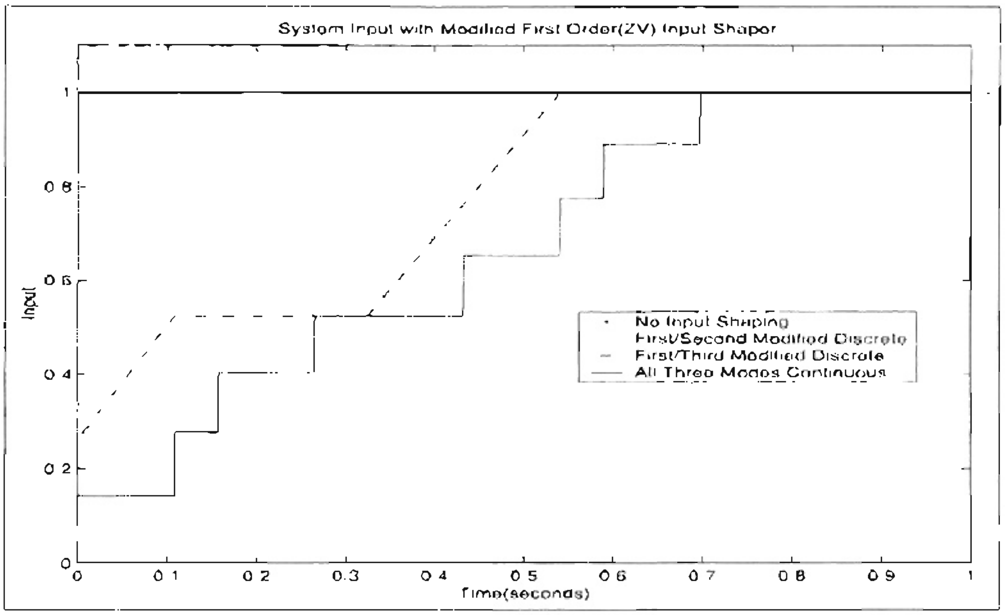

Figure 4.5: Modified First Order Shaper Inputs

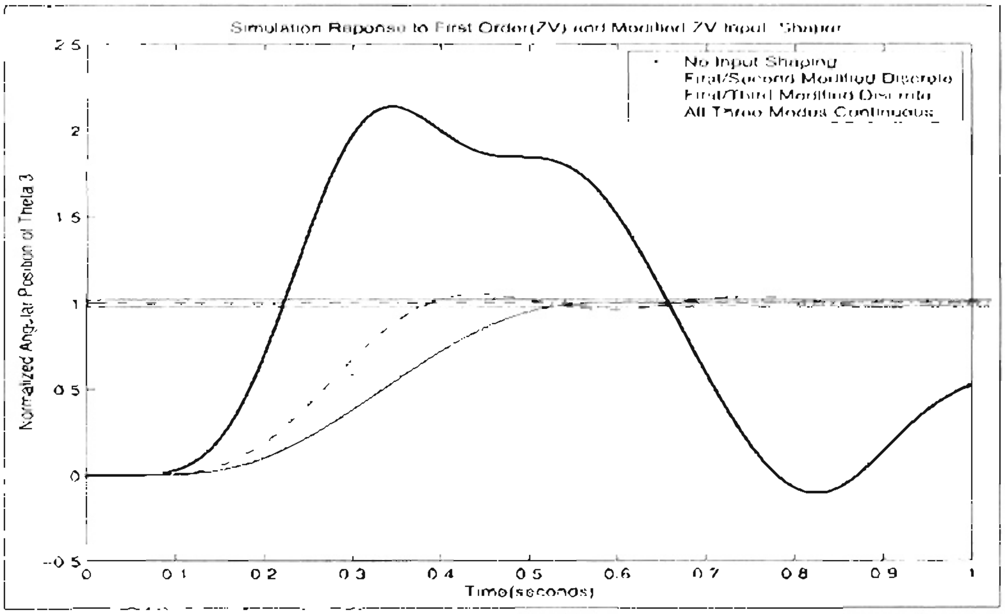

Figure 4.6: Modified First Order Shaper Step Response

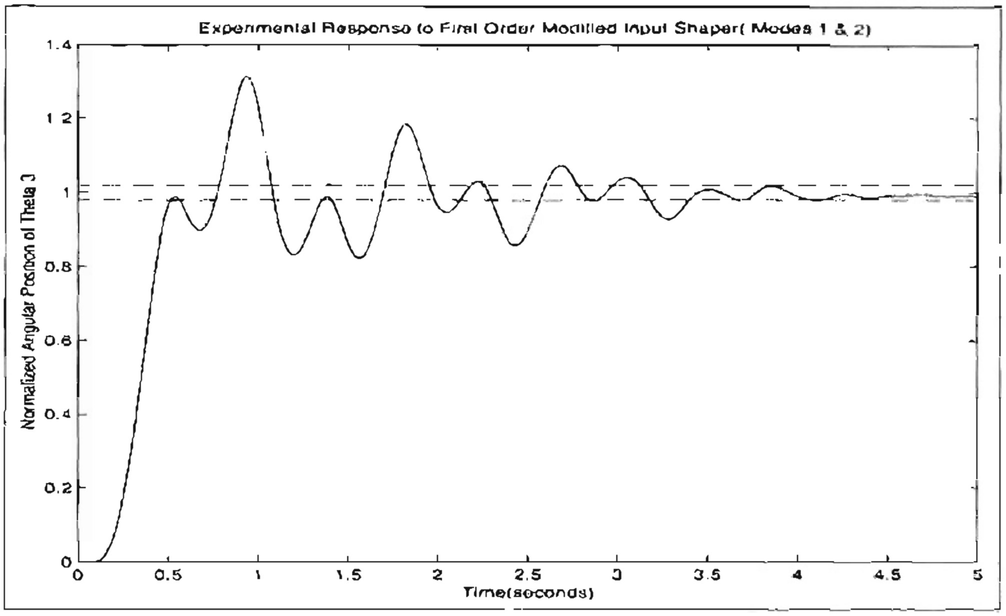

Figure 4.7: Modified(Mode 1 & 2) First Order Experimental Response

passing the un-shaped command signal through each filter is equivalent. Simulation and experimental results are shown in figure 4.6 and figure 4.7, respectfully.

The derivation of the input shaper for modes 1 and 3 is similar to the previous case. The sampling period for this case is the impulse time for the modified third mode.

$$T = m\Delta T_{32} \tag{4.25}$$

$$\therefore 4T = \Delta T_{12} \tag{4.26}$$

Now the input shaper equations for modes 1 and 3 can be found using the same sampling period.

$$u_1(t) = A_{11}\delta(t) + A_{12}\delta(t - \Delta T_{12}) = A_{11}\delta(t) + A_{12}\delta(t - 4T) \tag{4.27}$$

$$U_1(s) = A_{11} + A_{12}e^{-4Ts} \tag{4.28}$$

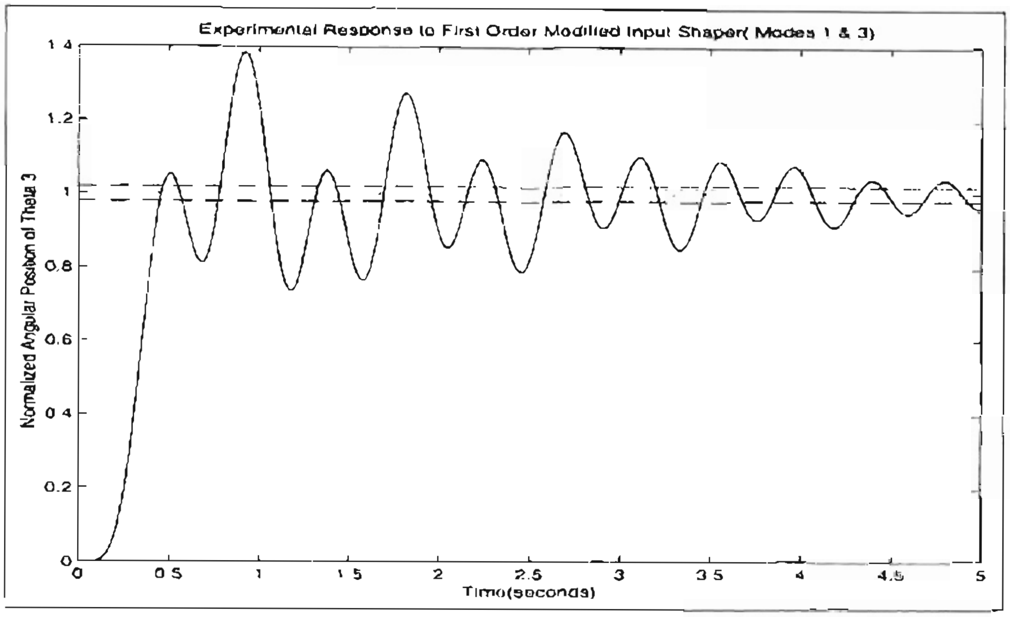

Figure 4.8: Modified(Mode 1 & 3) First Order Experimental Response

$$U_1(z) = A_{11} + A_{12}z^{-4} = \frac{A_{11}z^4 + A_{12}}{z^4} \tag{4.29}$$

$$mu_3(t) = mA_{31}\delta(t) + mA_{32}\delta(t - m\Delta T_{32}) = mA_{31}\delta(t) + mA_{32}\delta(t - T) \tag{4.30}$$

$$mU_3(s) = mA_{31} + mA_{32}e^{-Ts} \tag{4.31}$$

$$mU_3(z) = mA_{31} + mA_{32}z^{-1} = \frac{mA_{31}z + mA_{32}}{z} \tag{4.32}$$

Figure 4.6 also shows the simulation results for the first and third modified mode input shaper. Figure 4.8 shows the experimental results for this case. Figures 4.7 and 4.8 show that designing for the first two modes results in less residual vibration. This result is expected since the first two modes dominate the response of the system with respect to the third mode.

4.4 Digital Two Mode Second Order Shaper

The derivation of the modified second order input shaper is identical to the first order case with a couple of exceptions. An additional impulse is added to each impulse sequence and the time duration of each sequence is doubled.

The second order input shapers for the first mode and the modified second mode are defined as:

$$U_1(z) = A_{11} + A_{12}z^{-3} + A_{13}z^{-6} = \frac{A_{11}z^6 + A_{12}z^3 + A_{13}}{z^6} \tag{4.33}$$

$$mU_2(z) = mA_{21} + mA_{22}z^{-1} + mA_{23}z^{-2} = \frac{mA_{21}z^2 + mA_{22}z + mA_{23}}{z^2} \tag{4.34}$$

The second order input shapers for the first mode and the modified third mode are defined as:

$$U_1(z) = A_{11} + A_{12}z^{-4} + A_{13}z^{-8} = \frac{A_{11}z^8 + A_{12}z^4 + A_{13}}{z^8} \tag{4.35}$$

$$mU_3(z) = mA_{31} + mA_{32}z^{-1} + mA_{33}z^{-2} = \frac{mA_{31}z^2 + mA_{32}z + mA_{33}}{z^2} \tag{4.36}$$

The input for both cases is shown in figure 4.9 along with the un-shaped input and the input designed from Singer's continuous time technique. Figure 4.10 shows the system response for the inputs shown in figure 4.9. Experimental results for the 1/2 case and 1/3 case are shown in figure 4.11 and 4.12, respectfully.

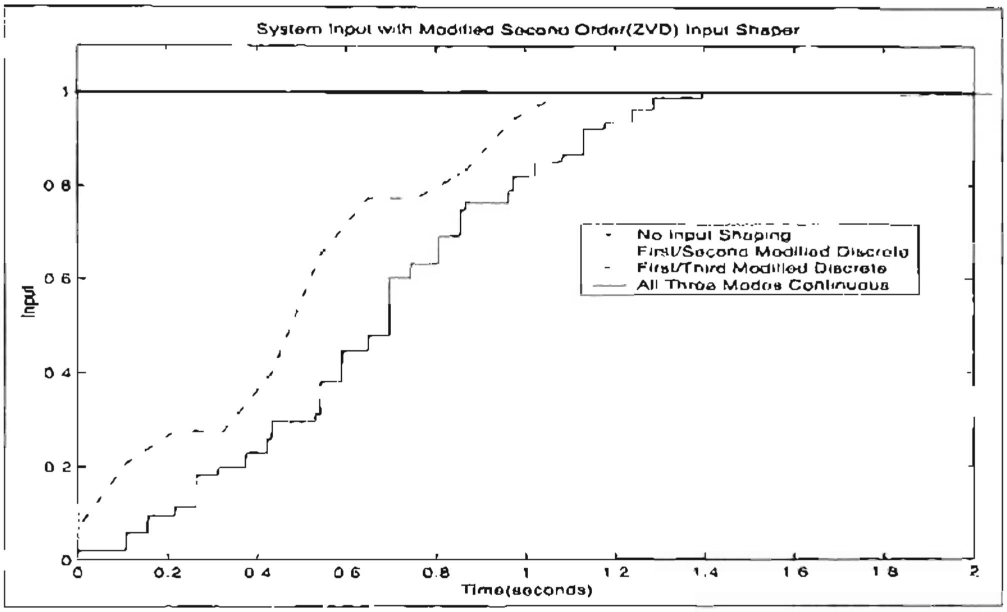

Figure 4.9: Modified Second Order Shaper Inputs

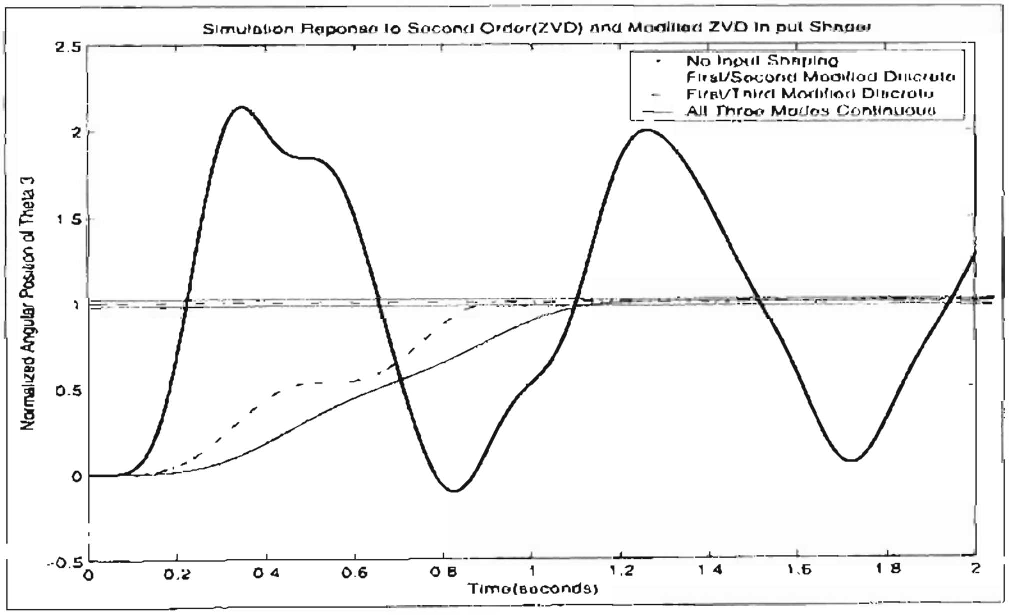

Figure 4.10: Modified Second Order Shaper Step Response

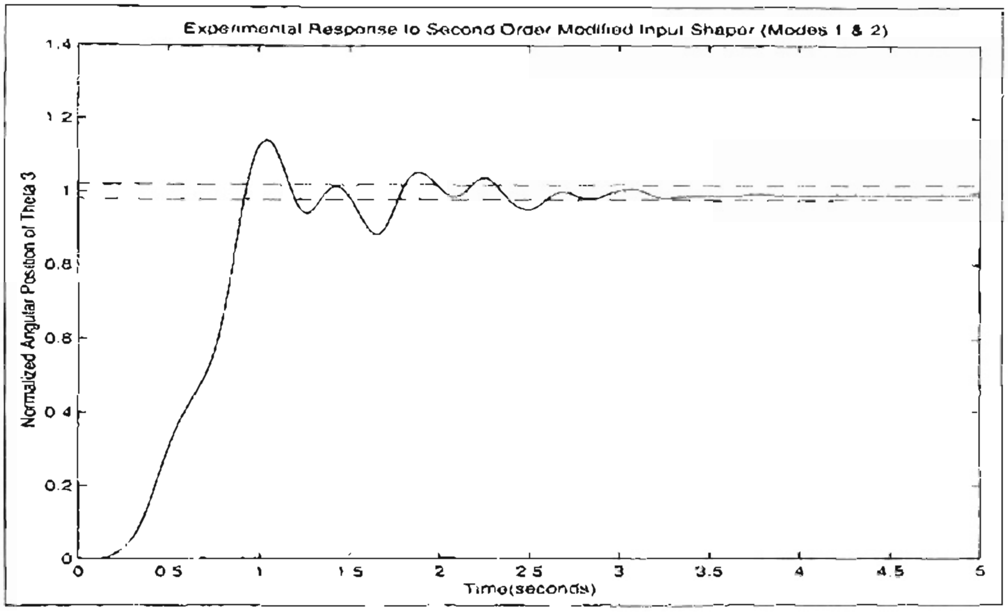

Figure 4.11: Modified(Mode 1 & 2) Second Order Experimental Response

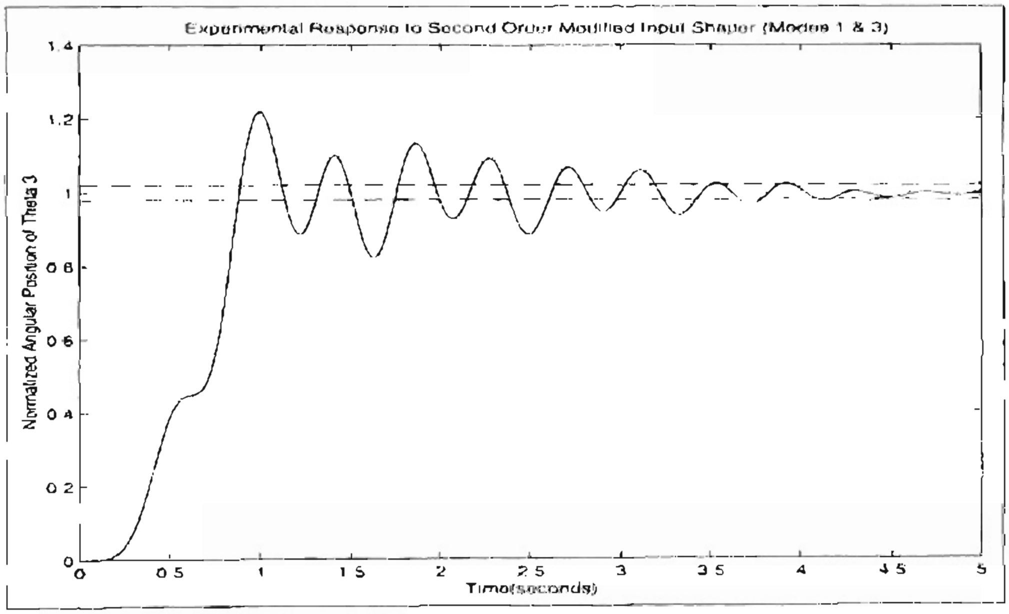

Figure 4.12: Modified(Mode 1 & 3) Second Order Experimental Response

Chapter 5

Discrete Time Zero Placement Technique

Using Tuttle's technique allows for the design of single or multiple mode input shapers with relative ease. Since the torsional plant is multiple mode system all input shapers designed in this chapter will be multiple mode input shapers. The shapers will vary in order of robustness selected for a particular mode. Before any shapers can be derived the input shaper zeros must be determined using the system's natural frequencies and damping ratios (ω_{n1}, ζ_1, ω_{n2}, ζ_2, ω_{n3},ζ_3). To avoid confusion, the variable for un-damped natural frequency changes because this technique uses both the un-damped and damped natural frequency.

$$p_1 = e^{-\zeta_1\omega_{n1}T}e^{j\omega_{d1}T} \tag{5.1}$$

$$p_1^* = e^{-\zeta_1\omega_{n1}T}e^{-j\omega_{d1}T} \tag{5.2}$$

$$p_2 = e^{-\zeta_2\omega_{n2}T}e^{j\omega_{d2}T} \tag{5.3}$$

$$p_2^* = e^{-\zeta_2 \omega_{n2} T} e^{-j\omega_{d2} T} \qquad (5.4)$$

$$p_3 = e^{-\zeta_3 \omega_{n3} T} e^{j\omega_{d3} T} \qquad (5.5)$$

$$p_3^* = e^{-\zeta_3 \omega_{n3} T} e^{-j\omega_{d3} T} \qquad (5.6)$$

The sampling period(impulse spacing), T, of the input shaper will be determined depending on the order of the input shaper, the desired impulse amplitudes, and the maximum time duration of the input shaper(if specified). The total time duration is a function of the sampling period and the number of impulses. The user must determine how much time delay in the system response is acceptable. Each subsequent section will present the derivation of a particular order shaper, simulation results, and experimental results.

5.1 Derivation of First Order Input Shaper

A first order discrete time shaper is derived by placing one pair of complex conjugate shaper zeros at the system poles. The shaper zeros are found using equations 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6. The discrete representation of a first order input shaper for all three modes is defined by equation 5.7. The shaper is not completely defined because the sampling period has not been selected.

$$H(z) = \frac{C(z-p_1)(z-p_1^*)(z-p_2)(z-p_2^*)(z-p_3)(z-p_3^*)}{z^6} \qquad (5.7)$$

The sampling period cannot be determined in the form that the shaper is represented in equation 5.7. Several steps will be outlined to find the sampling time, which determines the impulse amplitudes and times. First, the individual polynomials must be expanded to find the coefficients of the input shaper. Expanding the first order polynomials in equation 5.7 results in a sixth order polynomial that compensates for all three modes shown in equation 5.8.

$$H(z) = \frac{C(z^6 + a_1 z^5 + a_2 z^4 + a_3 z^3 + a_4 z^2 + a_5 z + a_6)}{z^6} \qquad (5.8)$$

The discrete transfer function in equation 5.8 can be represented as a polynomial by dividing the numerator with the denominator. The result is shown in equation 5.9.

$$H(z) = C(z + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4} + a_5 z^{-5} + a_6 z^{-6}) \qquad (5.9)$$

The polynomial expression in equation 5.9 is mapped to the Laplace domain using $z = e^{Ts}$ and the result is shown in equation 5.10.

$$h(t) = C[1 + a_1 e^{-Ts} + a_2 e^{-2Ts} + a_3 e^{-3Ts} + a_4 e^{-4Ts} + a_5 e^{-5Ts} + a_6 e^{-6Ts}] \qquad (5.10)$$

To find the continuous time representation of the input shaper the inverse Laplace transform of equation 5.10 is found. The result is shown by equation 5.11. The input shaper impulses are equally spaced over integer multiples of the input shaper sampling period, T, and the first impulse at time equal zero.

$$h(t) = C \begin{bmatrix} \delta(t) + a_1\delta(t - T) + a_2\delta(t - 2T) + a_3\delta(t - 3T) \\ + a_4\delta(t - 4T) + a_5\delta(t - 5T) + a_6\delta(t - 6T) \end{bmatrix} \qquad (5.11)$$

The coefficient, C, is determined when the sampling period and the impulse amplitudes are defined. For this case C is defined by equation 5.12:

$$C = (1 + a_1 + a_2 + a_3 + a_4 + a_5 + a_6)^{-1} \qquad (5.12)$$

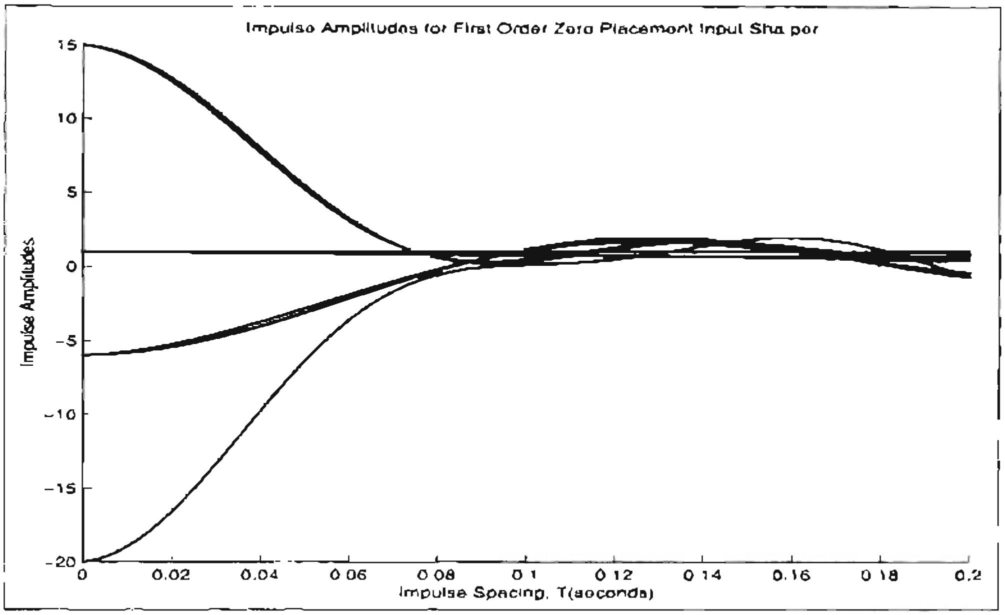

Figure 5.1: First Order Zero Placement Shaper Amplitudes

The sampling time, impulse amplitudes, time duration, and normalization coefficient are determined by plotting the impulse amplitudes versus the sampling period. The sampling period can be selected as the shortest sampling period that yields all positive impulses or it can be any sampling period that yields all positive impulses. Figure 5.1 shows the impulse amplitudes for the closed loop discrete time plant for an arbitrary range of T.

Using figure 5.1, T is selected to be 0.125 for simulations and experiments. A sampling period smaller than 0.125 would yield all positive impulse amplitudes, but experiments have shown that less aggressive(larger impulse spacing) input shapers have better performance. The simulation response and experimental response for $T = 0.125$ are shown in figures 5.2 and 5.3, respectfully.

A first order discrete time input shaper can completely eliminate residual vibration in simulation but still causes vibration experimentally. Shapers will be derived to reduce the level of residual vibration.

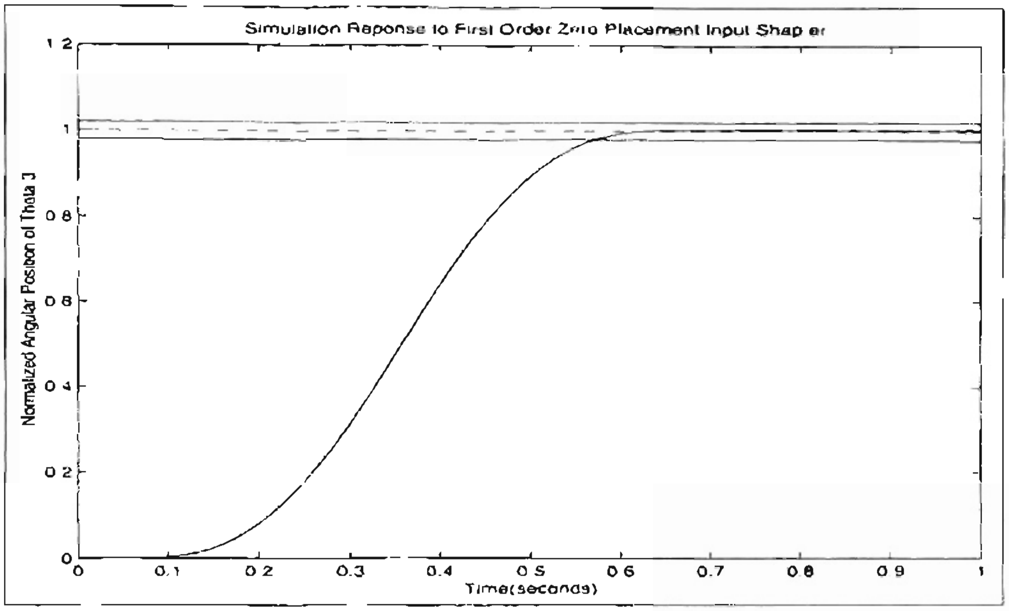

Figure 5.2: First Order Zero Placement Shaper Step Simulation Response

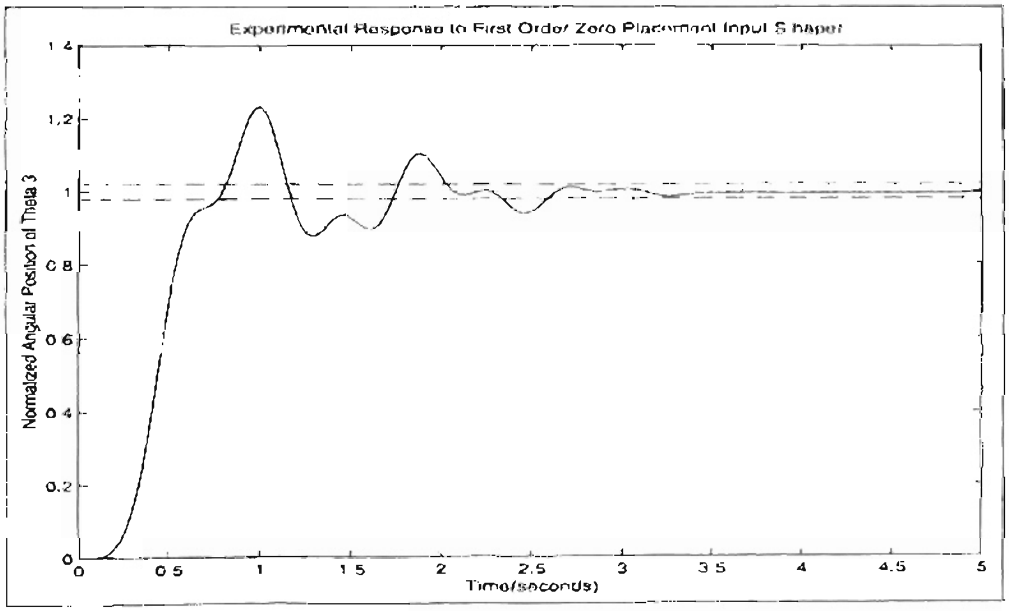

Figure 5.3: First Order Zero Placement Shaper Step Experimental Response

5.2 Derivation of Dominant Mode Robust Input Shaper

The slowest system mode contributes the largest amplitude of residual vibration in the system response. If all three modes are approximations and the first mode is the dominant mode, then the order of robustness of the portion of the shaper designed to eliminate the dynamics due to the first mode can be increased. This is done by adding another set of complex conjugate zeros on the system poles of the first mode(ζ_1 & ω_1).

$$H(z) = \frac{C(z-p_1)^2(z-p_1^*)^2(z-p_2)(z-p_2^*)(z-p_3)(z-p_3^*)}{z^8} \tag{5.13}$$

Equation 5.13 shows that adding shaper zeros only requires the power of a complex conjugate pair to be raised to the desired order. In this case the order for mode one is two or $n_1 = 2$, and the order for modes 2 and 3 is one or $n_2 = n_3 = 1$. The same procedure, as previously described, is used to find the input shaper sampling period and the impulse amplitudes. Using figure 5.4, the sampling period was selected as $T = 0.140$ seconds. Figure 5.5 shows the simulation response and figure 5.6 shows the experimental response of the "robust dominant mode" input shaper.

The "dominant mode" shaper reduce the level of residual vibration compared to the first order input shaper. A second order input shaper will be derived to investigate the cause of the residual vibration remaining in the response of the "dominant mode" input shaper.

5.3 Derivation of Second Order Input Shaper

A second order shaper for all three modes is designed to compensate for parameter variation of the dominant mode and the relative high frequency modes. Two input shaper zeros are placed at each system pole or $n_1 = n_2 = n_3 = 2$. This results in the shaper shown in equation 5.14.

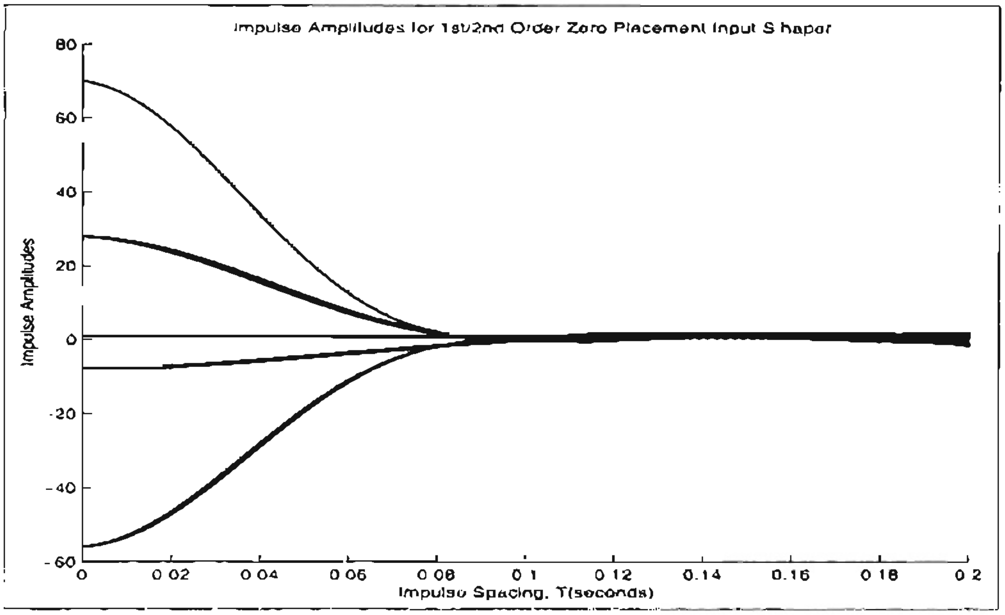

Figure 5.4: First/Second Order Zero Placement Shaper Amplitudes

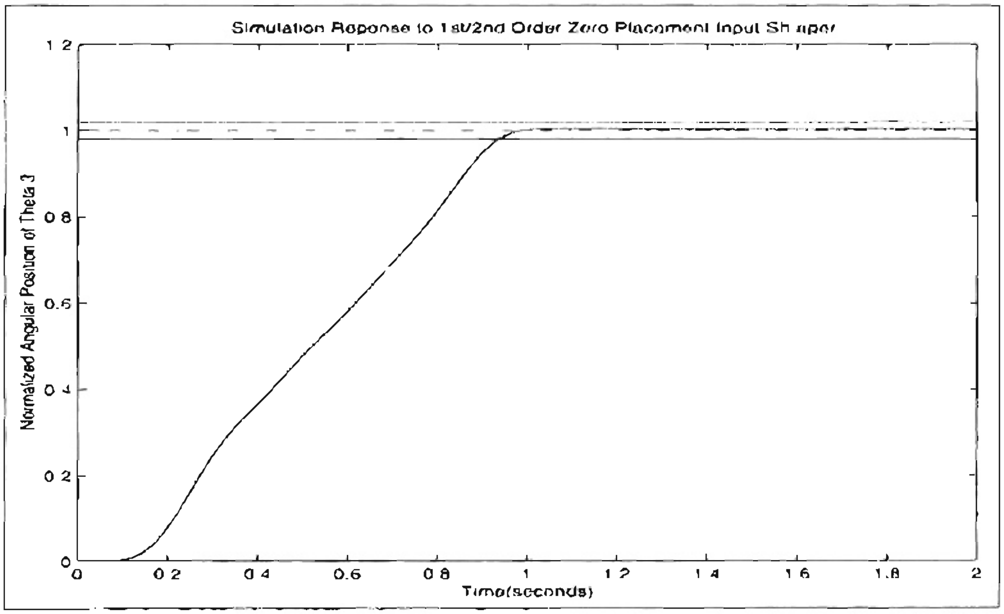

Figure 5.5: First/Second Order Zero Placement Shaper Step Simulation Response

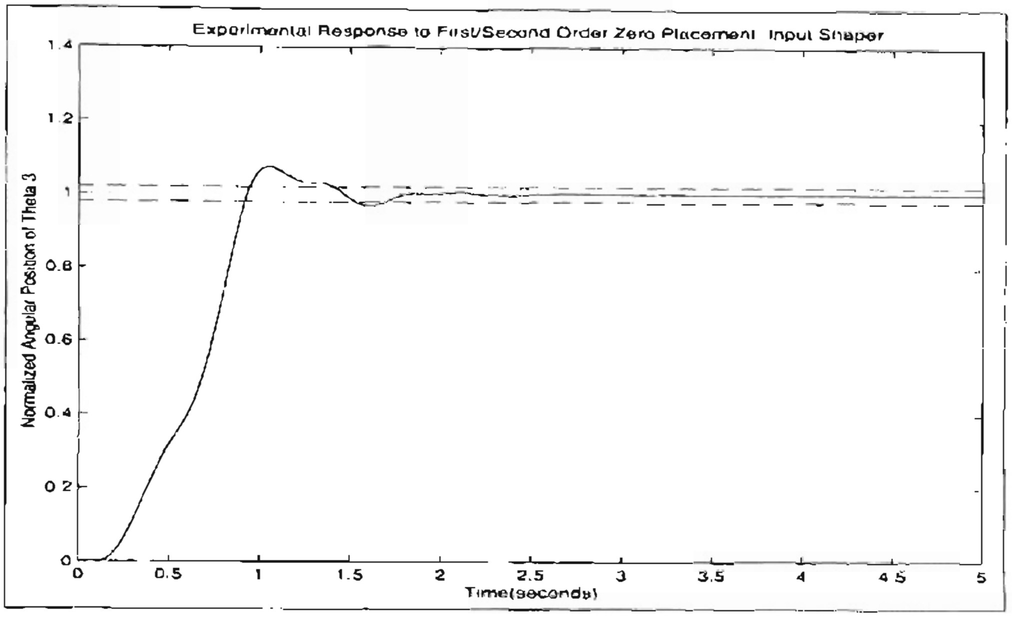

Figure 5.6: First/Second Order Zero Placement Shaper Step Experimental Response

$$H(z) = \frac{C(z-p_1)^2(z-p_1^*)^2(z-p_2)^2(z-p_2^*)^2(z-p_3)^2(z-p_3^*)^2}{z^{12}} \qquad (5.14)$$

Equation 5.14 shows that the causality constraint is still met. This is done by setting the power of the denominator equal to the overall order of the numerator or $r = 2(n_1 + n_2 + n_3)$. The closed form solution of equation 5.14 is not found, instead it is multiplied in MATLAB using *conv* for a range of sampling periods and plotted for that range. Figure 5.7 shows the impulse amplitudes for equation 5.14. The sampling period for this shaper is set to $T = 0.125$ seconds so a direct comparison can be made between the first order shaper and this second order shaper. The simulation results are shown in figure 5.8 and the experimental results by figure 5.9. Figure 5.9 shows that there is no reduction of residual vibration when compared to the "dominant mode" input shaper. This verifies that the first mode dominates the response and that if more reduction in residual vibration is required a more robust input shaper for the first mode must be implemented.

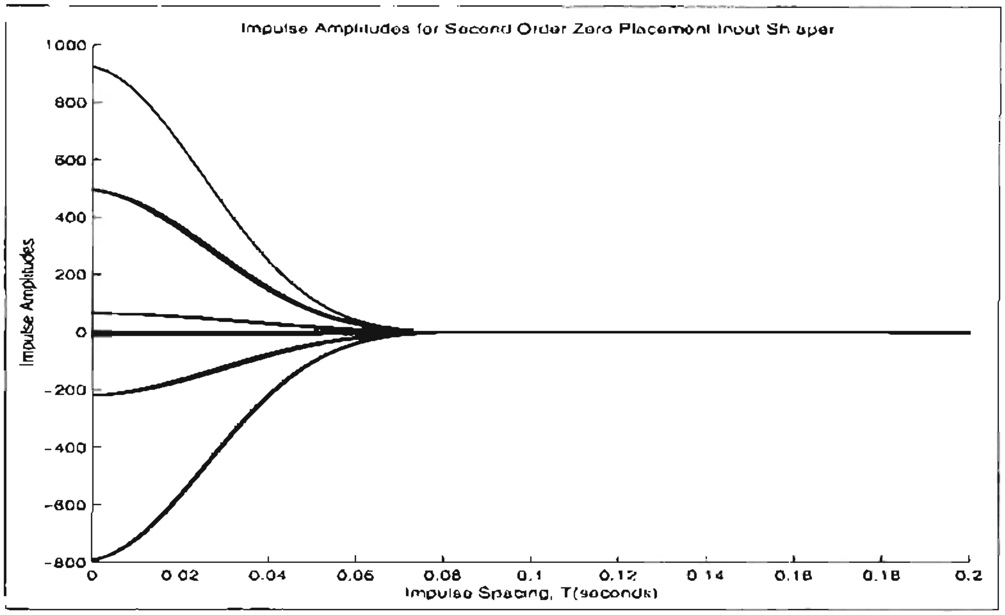

Figure 5.7: Second Order Zero Placement Shaper Amplitudes

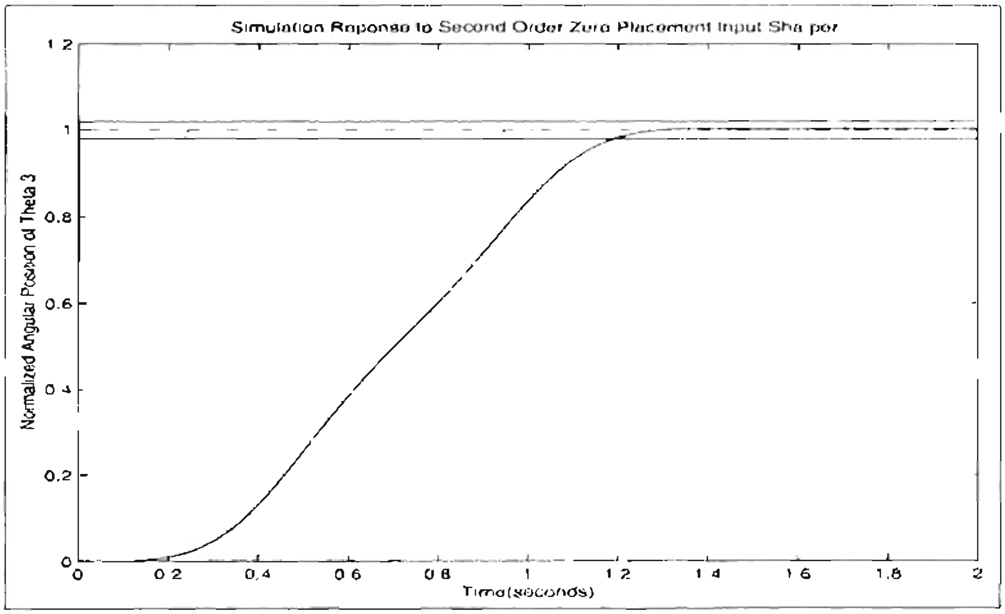

Figure 5.8: Second Order Zero Placement Shaper Step Simulation Response

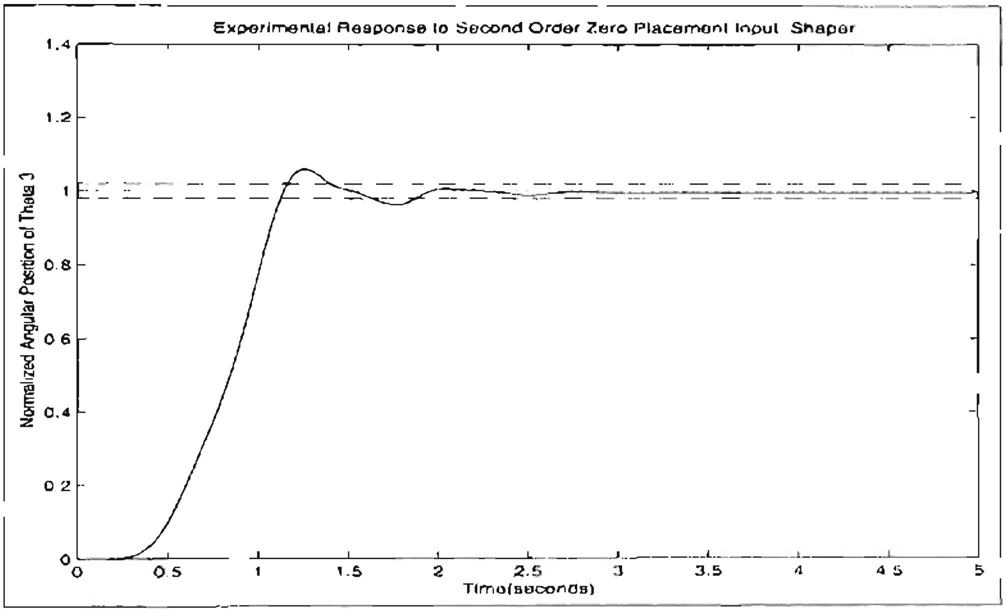

Figure 5.9: Second Order Zero Placement Shaper Step Experimental Response

Chapter 6

Extra-Insensitive and Negative Time-Optimal Input Shaping

This chapter outlines the derivation of several input shapers designed and implemented in continuous time. The shapers will again be convolved with a unit step input function to illustrate point-to-point control. Only simulation results will be shown for the techniques presented in this chapter.

6.1 Extra-Insensitive Input Shaping

The results of the previous input shaping techniques show that there is some parameter variation evident when first order(ZV) input shaping was implemented. Extra-Insensitive input shaping will be used to reduce residual vibration over a larger region of frequencies. Since One Hump EI input shapers have been shown to have the same time duration as a second order(ZVD) input shaper the results from the two input shapers will be compared.

A One Hump EI input shaper has three impulses and can be defined by equation 6.1. The level of residual vibration will be set to the maximum allowable level of $V = 5\%$.

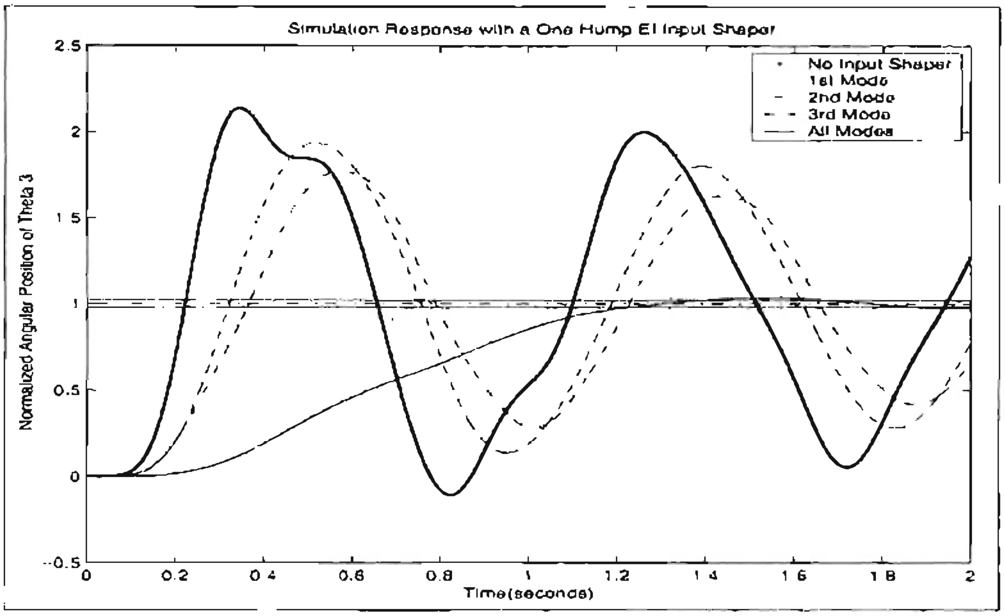

Figure 6.1: One Hump EI Shaper Simulation Response

$$u(t) = A_1\delta(t - T_1) + A_2\delta(t - T_2) + A_3\delta(t - T_3) \tag{6.1}$$

The impulse amplitudes and times are defined by the curve fit equations 2.49, 2.50, 2.51, 2.52, 2.53, and 2.54. The first impulse, A_1, is still implemented at T_1 0. The impulse sequence is determined by plugging the desired level of residual vibration and the system parameters into the equations for A_1, A_2, A_3, T_1, T_2, and T_3. The response for the closed loop system with PD control is shown in figure 6.1. The input shapers were found for each mode separately and then convolved for the multiple mode case using the same procedure presented in chapter 4. This technique cannot be implemented as a digital filter for either the single mode case or the multiple mode case because the impulse times are not equally spaced. Therefore, the robustness of this technique could not be verified experimentally.

6.2 Input Shapers with Positive and Negative Impulses

Several negative time optimal input shapers are derived in this section using Singhose's look up table method. The technique developed by Singhose, Seering, and Singer is limited to single mode design and has been shown to excite un-modeled high frequency dynamics. Therefore, an attempt is made to convolve negative time optimal input shapers in simulation.

First the single mode case is considered to study the effect of the un-modeled high frequency resonances on the system response to a unit step input. Both the Unity Magnitude and Partial Sum input shapers will be shown in this section. There are three impulses in a sequence for ZV negative "time-optimal" input shapers and the impulse sequence is defined by equation 6.2.

$$u(t) = A_1\delta(t - T_1) + A_2\delta(t - T_2) + A_3\delta(t - T_3) \tag{6.2}$$

Table 2.4 shows the curve fit solution for the unity magnitude and partial sum techniques. Both the zero-vibration(ZV) and zero-vibration-derivative(ZVD) are shown in table 2.4. The ZVD solution adds two more impulses to the input shaping sequence and it is represented by equation 6.3.

$$u(t) = A_1\delta(t - T_1) + A_2\delta(t - T_2) + A_3\delta(t - T_3) + A_4\delta(t - T_4) + A_5\delta(t - T_5) \tag{6.3}$$

The unity magnitude or partial sum input shapers are determined by plugging the system parameters into table 2.4 and for the input shaper with either equation 6.2 or 6.3. Only the ZV shapers are considered in this section. Again the shapers are derived for each mode and convolved to find the multiple mode solution. Figure 6.2 shows the system response for the UM-ZV shaper and figure 6.3 shows the response for the PS-ZV shaper. Both figures show that the single mode shaper for any mode results in a large level of residual vibration.

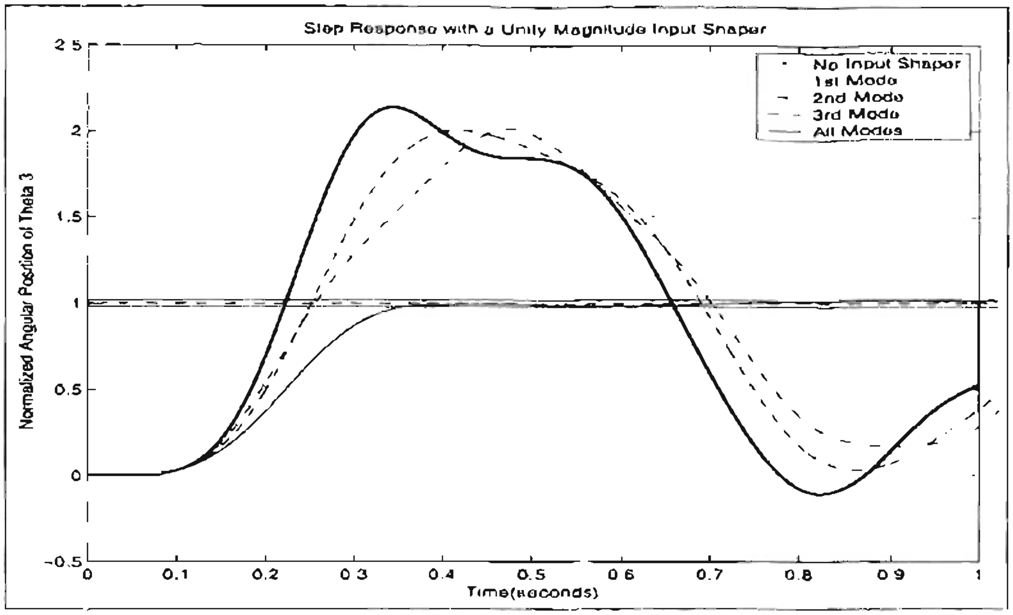

Figure 6.2: UM-ZV Shaper Simulation Response

When the input shapers are convolved residual vibration can be eliminated but the inputs for multiple mode UM-ZV and PS-ZV shapers will exceed actuator limits in a real system. Violation of actuator constraint is caused by a shaped input that exceeds the desired output for the UM-ZV shaper and impulse amplitudes that are greater than 1 and less than -1 for the PS-ZV shaper. The UM-ZV and PS-ZV input shapers could not be evaluated experimentally for these reasons.

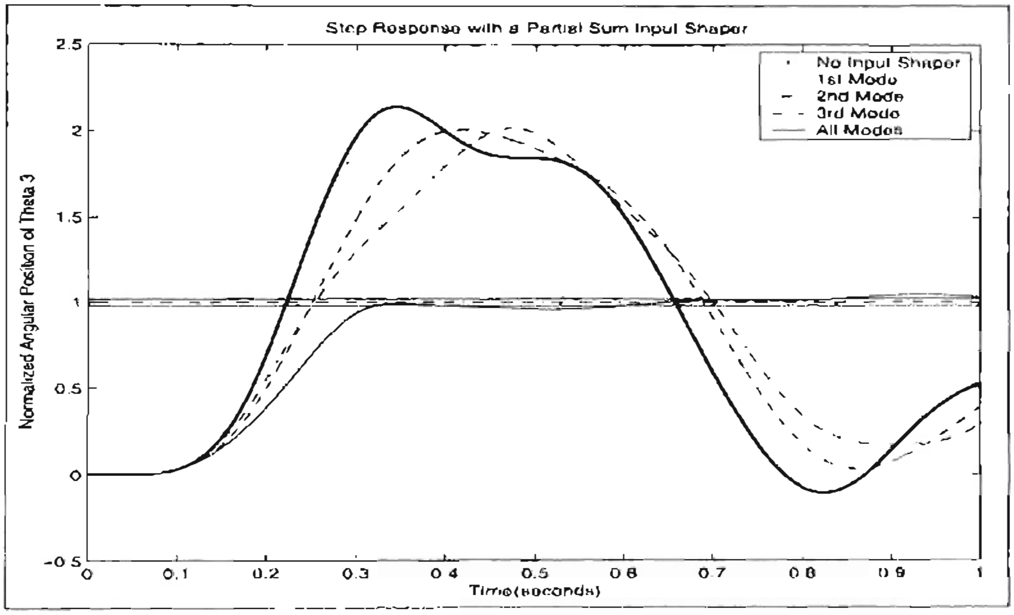

Figure 6.3: PS-ZV Shaper Simulation Response

Chapter 7

Implementation Method

The ECP Executive software package was used to implement the shaped inputs on the hardware. It allows easy changes in the input shape and duration, and also control algorithms that can be easily tuned. The ECP Executive offers both a continuous time control algorithm and a discrete time control algorithm. Since computer controlled machines are not continuous, the digital control algorithm was the most realistic choice for designing a controller. The PD controller used in simulation and experiments is shown in equation 7.1.

$$G_c(z) = (K_p + K_d) - K_d z^{-1} \qquad (7.1)$$

In simulation, setting the gains is straightforward but in the ECP Executive there are a couple of ways to implement the control. The first way is to set the gains in the PID dialog box provided by the ECP Executive. The second way is to use the general form of the control algorithm and set the gains of a discrete polynomial, where $E_1 = K_p + K_d$ and $E_2 = -K_d$ for equation 7.2.

$$E(z) = E_1 + E_2 z^{-1} \qquad (7.2)$$

For either method the desired feedback(encoder 1, 2, or 3) should be selected and the sampling period, T_s, should be selected for the control algorithm. Every discrete polynomial in the general form of the control algorithm is executed at the selected sampling period. This became a problem when input shapers were implemented on the Torsional Plant as a discrete filter because the sampling rate has to be set to the impulse spacing for the input shaper. If the sampling period is not set to the input shaping filter then the delays for the impulse sequence will be incorrect making the input shaper ineffective. When the sampling period was set to the filter delay time, the system went unstable when the control algorithm was implemented. All the delays for each shaper were greater than the minimum sampling rate, $T_s = 0.884ms$. Therefore, none of the techniques presented were implemented experimentally as a digital filter. The cause of this issue is not known at this time.

The solution to the digital timing issue was to use the user defined trajectory option in the ECP Executive program. Trajectory files are in text format but saved with a ".trj" extension. The first number in a trajectory file defines the number of points in the trajectory. The following numbers define the trajectory in either counts, radians, or degrees. The software allows the user to select the units for either user defined inputs or ECP Executive inputs. An example trajectory file is shown below:

```
5
1
2
3
4
5
```

The number of points in the sample trajectory file is five and the trajectory moves from zero to five. The time interval of these move times is the segment time selected in the user defined trajectory dialog box. The segment times can be selected to within $1ms$ of user's desired time interval. Inputs become less accurate as the segment time approaches the minimum sampling period, $T_s = 0.884ms$. Also, trajectory files can have no more than 275 points. This prevented any of the input shaping techniques designed in continuous time from being tested experimentally. Input shapers designed in simulation as digital filters could be implemented on the ECP hardware because the segment time was equal to the filter delay time. The filter delay time was always much larger than the minimum sampling period.

The output of the input shapers in simulations are used to create the trajectory file and the segment time is set to the impulse time spacing for input shapers that have equal time steps. The trajectory files were created in MATLAB using the "File input/output" functions. Below is an example of how to create a trajectory file in MATLAB:

```
MAG=input('Enter the Magnitude of the Input')
fid=fopen('Sample.trj','w');
length=size(u);
fprintf(fid,'%f\r ',[length(1,1);MAG*u(1:1:length(1,1))]);
fclose(fid);
```

The *input* command prompts the user to enter the desired magnitude of the input at the MATLAB command window. To create a file the *fopen* command writes a file called *Sample.trj*, with write permissions('*w*'), in the present working directory and creates the file identifier, *fid*. The number of points is determined by the length of the input created in SIMULINK and *length=size(u)* finds the length of the input, *u*. It is the user's responsibility to verify that the number of points does not exceed 275. The data is written to the file by *fprint*, where *fid* identifies the file. The data is written as floating point numbers, *%f*, separated by return characters, *r*. The *fclose* command closes the file identifier assigned to the trajectory file. All experimental data shown was generated using trajectory filse created with "File input/output" functions in MATLAB.

Chapter 8

Comparison of Techniques

This chapter will cover the factors that determine the performance of the input shapers designed in chapters 4, 5, and 6. It will also provide comparison tables so that equivalent shapers can be evaluated and discuss some of possible causes for differences in performance. It should be noted that the solutions that Singhose(chapter 6) derived for Extra Insensitive and Negative "Time-Optimal" input shapers are curve fit approximations. Table 8.1 outlines the criteria used to compare and rate the input shapers.

For comparison purposes, it is necessary to formulate a list of the input shapers designed for the Torsional Plant.

1. First Order(ZV) Singer Input Shaper
2. Second Order(ZVD) Singer Input Shaper
3. Discrete Time Modified Input Shaper Using Singer First Order Constraints
4. Discrete Time Modified Input Shaper Using Singer Second Order Constraints
5. First Order Discrete Time Shaper Using Zero Placement for All Three Modes
6. First(All Modes)/Second(Mode 1) Order Discrete Time Shaper Using Zero Placement

Criteria
Total Time(Duration)
Number of Impulses
Order or Level of Robustness
Calculation Difficulty
Digital or Continuous Time Derivation
Single or Multiple Mode Capability
Relative Performance on ECP Hardware

Table 8.1: Criteria for Comparison of Input Shapers

7. Second Order Discrete Time Shaper Using Zero Placement for All Three Modes

8. Unity Magnitude Input Shaper for Each Mode Individually and All Three Modes Combined

9. Partial Sum Input Shaper for Each Mode Individually and All Three Modes Combined

10. Extra-Insensitive Input Shaper for Each Mode Individually and All Three Modes Combined

With the exception of Tuttle's technique outlined in chapter 5, all the input shapers were designed in continuous time. For a single mode, it is easy to implement an input shaper designed in continuous time as a digital filter using Singer's method. This works because a single mode input shaper of any order designed with that method has equally spaced time intervals. The time intervals determine the sampling period of the digital filter. Section 4.4 showed that if the parameters of non-dominant modes are altered slightly such that individually designed shapers have a time interval that is an integer multiple of the fastest input(shortest duration) time delay, then the multiple mode input shaper can be implemented as a digital filter. Also, there is no need to convolve the input shapers because they can be organized as a series of cascaded filters with the same sampling period.

Singhose's curve fits for Extra-Insensitive, Unity Magnitude, and Partial Sum input shapers are not designed to yield equally spaced time intervals. Therefore, they cannot be implemented as a digital filter even for a single mode. Also, Singhose's methods for Negative "Time-Optimal" input shapers are not designed to compensate for multiple modes in a system. Simulations did show that multiple mode input shapers with negative impulses can eliminate residual vibration but the plot of the input shows that there are numerous points where the impulse amplitudes exceed 1 and -1. This will violate actuator constraints if the un-shaped input saturates the input. Only impulse sequences with all positive impulses can guarantee not to violate actuator constraints given that the un-shaped input does not exceed actuator limits. The Extra-Insensitive input shapers can be used to compensate for multiple modes, but the continuous time input shaper must be transformed into the Laplace domain to eliminate the difficulty of convolution.

Tuttle's input shaping technique is derived in the discrete time domain for multiple modes, which makes it easy to implement since it is already a digital filter. Also, robustness is added by increasing the number of zeros used to cancel the system poles. Another significant trait is that it allows the user to select the desired sampling time for the filter. Therefore, the filter can be tuned to minimize the level of residual vibration. There is no need to convolve this shaper with another shaper since it is already designed for multiple modes and it also does not have to be convolved with the un-shaped input. The un-shaped input can be passed through the filter and the output of the filter is the shaped command signal.

Table 8.2 shows the simulation and experimental results of the listed techniques. Experimental data could not be acquired for the techniques implemented in continuous time in simulation. Both of Singer's methods used completely eliminate residual vibration in simulation but a second order shaper for all three modes results in an input shaper with

27 impulses. A second order input shaper using Tuttle's discrete time technique only has 13 impulses, and is shown to eliminate residual vibration in simulation and reduce it experimentally. In general, input shapers with fewer impulses are easier to implement. A first order input shaper using Tuttle's method shows that parameter uncertainty exist because a shaped input using this method resulted in a settling time 2 seconds slower than the simulation results. Results from Tuttle's second order input shaper and $1^{st}/2^{nd}$ order input shaper show that adding robustness to the first mode results in a faster rise time. They also show that adding robustness to the second and third mode did not improve the settling time. The modified method for multiple modes shows that if Singer's method is altered then it can be implemented as a digital filter and it results in fewer impulses. The modified shapers for the first and second mode improve the settling time, but don't perform as well as Tuttle's method. Designing a shaper for the first and third mode showed that ignoring the second mode will result in a larger level of residual vibration than when the third mode is ignored. Sinhose's One Hump EI shaper allows for some residual vibration at the system parameters and it results in a slower settling time. Singhose's negative time optimal input shapers were implemented for all three modes in simulation and showed that they can significantly reduce residual vibration. They would however cause the limits of the actuators to be exceeded.

Tuttle's method and the modified method were the only ones that could be experimentally tested. Both improved the system response but Tuttle's method resulted in better performance than the modified method.

Comparison Table

Input Shaping Method	Shaper Time Duration	Number Of Impulse	Sim. Rise Time	Sim. Settling Time	Exp. Rise Time	Exp. Settling Time
Singer First Order	0.6974	8	0.575	0.528	N/A[1]	N/A[1]
Singer Second Order	1.3948	27	1.210	1.126	N/A[1]	N/A[1]
Singhose One Hump EI	1.3948	27	1.306	3.368	N/A[1]	N/A[1]
Singhose UM-ZV	0.4650	27	0.718	0.358	N/A[2]	N/A[2]
Singhose PS-ZV	0.4086	27	0.646	3.563	N/A[2]	N/A[2]
Tuttle First Order	0.75	7	0.624	0.571	0.794	2.591
Tuttle Second Order	1.50	13	1.290	1.192	1.137	1.875
Tuttle 1st/2nd Order	1.12	9	0.984	0.935	0.929	1.695
Mod.(1/2) First Order	0.5764	5	0.440	2.031	0.772	3.401
Mod.(1/3) First Order	0.5404	6	0.387	2.006	0.458	N/A[3]
Mod.(1/2) Second Order	1.1528	9	1.042	0.985	0.925	2.59
Mod.(1/3) Second Order	1.0807	11	0.920	0.878	0.880	4.166

Table 8.2: Table of Simulation and Experimental Results

[1] Input Trajectory Files Exceeded 275 Points because of Time Step Limitations

[2] Input Trajectory Files Exceeded 275 Points because of Time Step Limitations and Actuator Limit Exceeded by Shaped Input Trajectory File

[3] Experimental Settling Time Greater than 5 Seconds

Chapter 9

Conclusions

Several previously developed input shaping techniques have been outlined and implemented on the Torsional Plant. Simulations verify that input shaping in any form can completely eliminate all residual vibration from a system response to a step input. If an accurate model is developed then low order or robust shapers can be effective. Experiments showed that the mathematical model derived in chapter 3 is not exact and that there must be some variations in the system's parameters probably due to un-modeled nonlinear dynamics. Shapers designed for more robust performance showed that the significance of the variations can be accounted for in the design of the impulse sequences. Experiments also showed that input shapers implemented as discrete filters are more easily implemented in the ECP Executive software because the impulse times are the sampling period of the filter. Some of the limitations of the software effected the accuracy to which the input shapers could be implemented.

9.1 Contributions

A new technique based on Singer's method was developed to implement input shapers designed in continuous time as digital filters for multiple mode systems. The motivation of this technique was that implementing input shapers as filters is simpler than the implementation of continuous time input shapers. As previously stated, the un-shaped command signal is passed through(multiplied) the digital filter and the output of the filter became the new command signal. This method eliminated the requirement of convolving input shapers to form a multiple mode input shaper. The digital filters were cascaded to perform the same task.

9.2 Future Work

The trajectory files were not implemented with 100% accuracy because the inputs have decimal remainders that the ECP Executive truncates to integer multiples of one count, since the encoders are not capable of reading in between two counts. Also, several input shapers required a small segment time relative to the minimum sampling period, T_s. This causes the impulse times to be inaccurately located. These two effects are detrimental to input shapers because the shaper times and amplitudes of a sequence are what reduces residual vibration.

All computer controlled machines will have similar limitations since it is not possible for them to be continuous. Therefore, the effects of quantization and digital timing on input shaping techniques are important topics for future research on input shaping.

Bibliography

[1] S.P. Bhat and D.K. Miu, "Solutions to Point-to-Point Control Problems Using Laplace Transform Technique," *Journal of Dynamic Systems, Measurement, and Control*, pp. 425-431, September 1991.

[2] Ray E. Bolz and George L. Tuve, "CRC Handbook of Tables for Applied Engineering Science," *CRC Press, Inc.*, Boca Raton, Florida, pp. 1071, 1973.

[3] J.M. Hyde and W.P. Seering, "Using Input Command Pre-Shaping to Suppress Multiple Mode Vibration," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, pp. 2604-2609, April 1991.

[4] L.Y. Pao and W.E. Singhose, "On the Equivalence of Minimum Time Input Shaping with Traditional Time-Optimal Control," *Proceedings of the IEEE Conference on Control Applications*, pp. 1120-1125, 1995.

[5] L.Y. Pao and W.E. Singhose, "A Comparison of Constant and Variable Amplitude Command Shaping Techniques for Vibration Reduction," *Proceedings of the IEEE Conference on Control Applications*, pp. 875-881, 1995.

[6] L.Y. Pao, T.N. Chang, and E. Hou, "Input Shaper Designs for Minimizing the Expected Level of Residual Vibration in Flexible Structures," *Proceedings of the American Control Conference*, Albuquerque, New Mexico, pp. 3542-3546, June 1997.

[7] L.Y. Pao and M.A. Lau, "Input Shaping Designs to Account for Uncertainty in Both Frequency and Damping in Flexible Structures," *Proceedings of the American Control Conference*, Philadelphia, Pennsylvania, pp. 3070-3071, June 1998.

[8] L.Y. Pao and C.F. Cutforth, "An Anaysis and Comparison of Frequency-Domain and Time-Domain Input Shaping," *Proceedings of the American Control Conference*, Philadelphia, Pennsylvania, pp. 3072-3074, June 1998.

[9] L.Y. Pao, "Multi-Input Shaping Design for Vibration Reduction," *Automatica*, pp. 81-89, 1999.

[10] Thomas R. Parks, "Torsional Dynamic System Manual," *Educational Control Products*, Woodland Hills, California, 1996.

[11] B.W. Rappole, N.C. Singer, and W.P. Seering, "Input Shaping with Negative Sequences for Reducing Vibrations in Flexible Structures," *Proceedings of the American Control Conference*, San Francisco, California, pp. 2695-2699, June 1993.

[12] N.C. Singer, "Residual Vibration Reduction in Computer Controlled Machines," *PhD thesis, Department of Mechanical Engineering*, MIT. Fall, 1988.

[13] N.C. Singer and W.P. Seering, "Using Acausal Shaping Techniques to Reduce Robot Vibration," *International Conference on Robotics and Automation*, Philadelphia, PA, pp. 1434-1439, April 25-29, 1988.

[14] N.C. Singer and W.P. Seering, "Design and Comparison of Command Shaping Methods for Controlling Residual Vibration," *International Conference on Robotics and Automation*, pp. 888-893, 1989.

[15] N.C. Singer and W.P. Seering, "Preshaping Command Inputs to Reduce System Vibration," *Journal of Dynamic Systems, Measurement, and Control*, Cambridge, MA, pp. 76-82, March 1990.

[16] N.C. Singer and W.P. Seering, "Experimental Verification of Command Shaping Methods for Controlling Residual Vibration in Flexible Robots," *The 1990 American Control Conference* , San Diego, CA, pp. 1738-1744, May 23-25, 1990.

[17] N.C. Singer and W.P. Seering, "An Extension of Command Shaping Methods for Controlling Residual Vibration Using Frequency Sampling," *Proceedings of the 1992 International Conference on Robotics and Automation*, Nice, France, pp. 800-805, May 1992.

[18] N.C. Singer and W.P. Seering, "An Efficient Algorithm for the Generation of Multiple-Mode Input Shaping Sequences," *Proceedings of the 1996 IEEE International Conference on Control Applications*, Dearborn, MI, pp. 373-378, September 15-18, 1996.

[19] T. Singh and S.R.Vadali, "Robust Time-Delay Control of Multiplemode Systems," *Proceedings of the American Control Conference*, Baltimore, Maryland, pp. 681-685, June 1994.

[20] W.E. Singhose, "Command Generation For Flexible Systems," *PhD thesis, Department of Mechanical Engineering*, MIT, May 1997.

[21] W.E. Singhose, W.P. Seering, and N.C. Singer, "Shaping Inputs to Reduce Vibration: A Vector Diagram Approach," *IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, pp. 922-927, May 1990.

[22] W.E. Singhose, W.P. Seering, and N.C. Singer, "Residual Vibration Reduction Using Vector Diagrams to Generate Shaped Inputs," *Journal of Mechanical Design*, pp. 654-659, June 1994.

[23] W.E. Singhose and N.C. Singer, "Initial Investigations into the Effects of Input Shaping on Trajectory Following," *Proceedings of the American Control Conference*, Baltimore, Maryland, pp. 2526-2532, June 1994.

[24] W.E. Singhose and T. Chuang, "Reducing Deviations from Trajectory Components with Input Shaping," *Proceedings of the 1995 American Control Conference*, 1995.

[25] W.E. Singhose, W.P. Seering, and N.C. Singer, "Comparison of Command Shaping Methods for Reducing Residual Vibration," *Proceedings of the 1995 European Control Conference*, 1995.

[26] W.E. Singhose, L.J. Porter and N.C. Singer, "Vibration Reduction Using Multi-Hump Extra-Insensitive Input Shapers," *Proceedings of the American Control Conference*, Seattle, Washington, pp. 3830-3834, June 1995.

[27] W.E. Singhose and N.C. Singer, "Effects of Input Shaping on Two-Dimensional Trajectory Following," *IEEE Transactions on Robotics and Automation*, pp. 881-887, December 1996.

[28] W.E. Singhose, S. Derezinski, and N.C. Singer, "Extra-Insensitive Input Shapers for Controlling Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, pp. 385-391, March-April 1996.

[29] W.E. Singhose and L.Y. Pao, "Comparison of Input Shaping Techniques for Speed-Critical Multi-Mode Flexible Systems," *Proceedings of the 1996 IEEE Conference on Control Applications*, Dearborn, Michigan, pp. 379-384, September 15-18, 1996.

[30] W.E. Singhose, W.P. Seering, and N.C. Singer, "Time-Optimal Negative Input Shapers," *Journal of Dynamic Systems, Measurement, and Control*, pp. 198-205, October 19, 1993.

[31] W.E. Singhose, E. Crain, and W. Seering, "Convolved and Simultaneous Two-Mode Input Shapers," *IEE Proceedings. Control Theory and Applications*, pp. 515-520, November 1997.

[32] W.E. Singhose and Karen Grosser, "Limiting Excitation of Unmodeled High Modes With Negative Input Shapers," *Proceedings of the 1999 IEEE International Conference on Control Applications*, Kohala Coast-Island of Hawaii, Hawaii, pp. 545-550, August 22-27, 1999.

[33] W.E. Singhose and B. W. Mills, "Command Generation Using Specified-Negative-Ampitude Input Shapers," *Proceedings of the Control Conference*, San Diego, California, pp. 61-65, June 1999.

[34] O.J.M. Smith, "Feedback Control Systems," *McGraw-Hill Book Company, Inc.*, New York, 1958.

[35] T.D. Tuttle and W.P. Seering, "A Zero Placement Technique for Designing Shaped Inputs to Suppress Multiple-mode Vibration," *Proceedings of the American Control Conference*, Baltimore, MD, pp. 2533-2537, June 1994.

Appendix A

MATLAB Codes(Model Builder and Mode Finder)

A.1 model.m

```
%%%%%%%%%%%%%%%%% MODEL BUILDER  %%%%%%%%%%%%%%%%%


%%%%%%%%  3 DOF System  %%%%%%%%

% Transfer Function

J1=0.01063;J2=0.01063;J3=J2;

c1=0.027;c2=0.002;c3=c2;

k1=2.76;k2=2.76;k3=2.76;

N1=sf*[J2*J3 (J2*c3+J3*c2)J2*k3+J3*(k2+k3)+c2*c3)

(c2*k3+c3*(k2+k3)),k2*k3];

N2=sf*k2*[J3 c3 k3];

N3=sf*k2*k3;

N=[N1;[0 0 N2];[0 0 0 0 N3]];


D=[J1*J2*J3,(J1*J2*c3+J1*J3*c2+J2*J3*c1),...
```

```
   (J1*(J2*k3+J3*(k2+k3)+c2*c3)...
   +J2*(J3*(k1+k2)+c1*c3)+J3*c1*c2), ...
   (J1*(c2*k3+c3*(k2+k3))+J2*(c1*k3+c3*(k1+k2)) ...
   +J3*(c1*(k2+k3)+c2*(k1+k2))+c1*c2*c3), ...
   J1*k2*k3+J2*(k3*(k1+k2))+J3*(k1*(k2+k3)+k2*k3) ...
   +c1*(c2*k3+c3*(k2+k3))+c2*c3*(k1+k2), ...
   c1*k2*k3+c2*k3*(k1+k2)+c3*(k1*(k2+k3)+k2*k3),...
   k1*k2*k3];

% State Space Model
A1=[0 1 0 0 0 0];
A2=[-(k1+k2)/J1 -c1/J1 k2/J1 0 0 0];
A3=[0 0 0 10 0];
A4=[k2/J2 0 -(k2+k3)/J2 -c2/J2 k3/J2 0];
A5=[0 0 0 0 0 1];
A6=[0 0 k3/J3 0 -k3/J3 -c3/J3];
A=[A1;A2;A3;A4;A5;A6];
B=[0 1/J1 0 0 0 0 ]';
C=[0 0 0 0 1 0];
DD=0;
```

A.2 mode.m

```
% The following finds the natural frequencies and mode
% shapes (eigenvalues and eigenvectors) of the system.
% You must first run "Model Builder" to obtain "A".
% Frequencies are in Hz.
% Mode shapes follow the same order as the Frequencies &
% are approximate because of non-proportional damping.

[U,wn]=eig(A);U=real(U);

%Calculate the Open Damping Ratios, d
[Wn,Z] = damp(ss(A,B,C,DD));

%Calculates the Frequencies in Hz
wn=abs(wn)/2/pi;

% Use the following for 3 DOF systems
Freq_Hz=[wn(1,1);wn(3,3);wn(5,5)]
Freq_RadPerS=Freq_Hz*2*pi
Modes=[[U(1,1);U(3,1);U(5,1)]/U(1,1),
[U(1,3);U(3,3);U(5,3)]/U(1,3), [U(1,5);
U(3,5);U(5,5)]/U(1,5)]
```

A.3 Discrete.m

```
%Find Discrete Time Model and Mode Shape
%Discrete.m
%Script That Creates the Model
model
%Simulink Model that is used in dlinmod
%Open Loop State Space PD Control Model
open OLSSPDModel
```

```
Ts=0.000884*int;%0.884ms is the minimum Ts
%Continuous Time Open Loop System Matrix
sysc=ss(A,B,C,DD);
%Discrete Time Open Loop System Matrix
sysd=c2d(sysc,Ts);
%Discrete Time State Space Matrix
[Ad,Bd,Cd,Dd,Ts] =ssdata(sysd);
%Open Loop State Space Model with PD Control
[Adt,Bdt,Cdt,Ddt]=dlinmod('OLSSPDModel',Ts);
%Open Loop System Matrix with PD Control
sysdt=ss(Adt,Bdt,Cdt,Ddt,Ts);
%Closed Loop System Matrix with PD Control
sysdtcl=feedback(sysdt,1);

%Determine the damping ratio and natural
%frequencies of the modes
[Wn,Z]=damp(sysdtcl);
```

Appendix B

MATLAB Codes(Singer's Method)

B.1 DisPlantFirstOrderCLPD.m

```
%1st Order Input Shaper for Closed Loop System

%With PD Control

%DisPlantFirstOrderCLPD.m


%Change to the Correct Directory

cd c:\Research\SingerShaper

%Clears All Old Variables

clear all, close all

%Clears Command Prompt

clc

%PD Control Gains

Kp=0.06;

Kd=0.75;

%Sampling Time of the ECP System

int=1; %Used To Specify Sampling

       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%1st Order Input Shaper for CL System for First Mode
%Damping Coefficient and System Natural Frequency
zeta1c = Z(1,1);
w1c=Wn(1,1);
K1c = exp(-zeta1c*pi/sqrt(1-zeta1c^2));
deltaT1c = pi/(w1c*sqrt(1-zeta1c^2));
%First Impulse at Time=0
A_11c = 1/(1+K1c);
%Second Impulse at Time=deltaT
A_12c = K1c/(1+K1c);
%1st Order Input Shaper for CL System for Second Mode
%Damping Coefficient and System Natural Frequency
zeta2c = Z(3,1);
w2c=Wn(3,1);
K2c = exp(-zeta2c*pi/sqrt(1-zeta2c^2));
deltaT2c = pi/(w2c*sqrt(1-zeta2c^2));
%First Impulse at Time=0
A_21c = 1/(1+K2c);
%Second Impulse at Time=deltaT
A_22c = K2c/(1+K2c);
%1st Order Input Shaper for CL System for Second Mode
%Damping Coefficient and System Natural Frequency
zeta3c = Z(5,1);
w3c=Wn(5,1);
```

```
K3c = exp(-zeta3c*pi/sqrt(1-zeta3c^2));
deltaT3c = pi/(w3c*sqrt(1-zeta3c^2));
%First Impulse at Time=0
A_31c = 1/(1+K3c);
%Second Impulse at Time=deltaT
A_32c = K3c/(1+K3c);
%Run DisPlantFirstOrderCLPDModel in Simulink
%This block diagram simulates a step input with
%input shaping for each individual mode
sim('DisPlantFirstOrderCLPDModel')
%Stead State value found for closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(t));
ss1=1.02*ones(size(t));
ss2=0.98*ones(size(t));
figure(1)
plot(t,yc/ss(temp(1,1)),'.',t,y1c/ss(temp(1,1)),':',...
t,y2c/ss(temp(1,1)),'-.',t,y3c/ss(temp(1,1)),'--',...
t,ycAllMode/ss(temp(1,1)),'-',t,ys,'k-.',t,ss1,'k',...
t,ss2,'k')
axis([0 1 -0.5 2.5])
title('Simulation Reponse to First Order(ZV) Input Shaper')
xlabel('Time(seconds)')
ylabel('Normalized Angular Position of Theta 3')
%grid
legend('No Input Shaping','First Mode',...
   'Second Mode','Third Mode','All Three Modes')
```

```
figure(2)
plot(t,uc,'.',t,u1c,':',t,u2c,'-.',t,u3c,'--',...
     t,ucAllMode,'-')
axis([0 1 0 1.1])
title('System Input with First Order(ZV) Input Shaper')
xlabel('Time(seconds)')
ylabel('Input')
%grid
legend('No Input Shaping','First Mode',...
   'Second Mode','Third Mode','All Three Modes',0)
%Create the Trajectory file for ECP Hardware Experiments
%NOTE:  SET THE SEGMENT TIME EQUAL TO Ts ms !!!!!!!!!!!!
MAG=input('Desired Number Counts: \n');
cd /, cd Research/ECPInputs
fid=fopen('DisPlantFirstOrderCLPD.trj','w');
length=size(ucAllMode);
fprintf(fid,'%f\r',[length(1,1);...
                    MAG*ucAllMode(1:1:length(1,1))]);
fclose(fid);
cd c:\Research\SingerShaper
```

B.2 DisPlantSecondOrderCLPD.m

```
%2nd Order Input Shaper for Closed Loop System
%With PD Control
%DisPlantSecondOrderCLPD.m
```

```
%Change to the Correct Directory
cd c:\Research\SingerShaper
%Clears All Old Variables
clear all, close all
%Clears Command Prompt
clc
%PD Control Gains
Kp=0.06;
Kd=0.75;
%Sampling Time of the ECP System
int=1; %Used To Specify Sampling
       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%1st Order Input Shaper for CL System for First Mode
%Damping Coefficient and System Natural Frequency
zeta1 = Z(1,1);
w1=Wn(1,1);
K1 = exp(-zeta1*pi/sqrt(1-zeta1^2));
T_12 = pi/(w1*sqrt(1-zeta1^2));
T_13 = 2*T_12;
%First Impulse at Time=0
A_11 = 1/(1+2*K1+K1^2);
%Second Impulse at Time=T_12
A_12 = 2*K1/(1+2*K1+K1^2);
%Third Impulse at Time=2*T_12
```

```
A_13 = (K1^2)/(1+2*K1+K1^2);
%1st Order Input Shaper for CL System for Second Mode
%Damping Coefficient and System Natural Frequency
zeta2 = Z(3,1);
w2=Wn(3,1);
K2 = exp(-zeta2*pi/sqrt(1-zeta2^2));
T_22 = pi/(w2*sqrt(1-zeta2^2));
T_23 = 2*T_22;
%First Impulse at Time=0
A_21 = 1/(1+2*K2+K2^2);
%Second Impulse at Time=T_22
A_22 = 2*K2/(1+2*K2+K2^2);
%Third Impulse at Time=2*T_22
A_23 = (K2^2)/(1+2*K2+K2^2);
%1st Order Input Shaper for CL System for Third Mode
%Damping Coefficient and System Natural Frequency
zeta3 = Z(5,1);
w3=Wn(5,1);
K3 = exp(-zeta3*pi/sqrt(1-zeta3^2));
T_32 = pi/(w3*sqrt(1-zeta3^2));
T_33 = 2*T_32;
%First Impulse at Time=0
A_31 = 1/(1+2*K3+K3^2);
%Second Impulse at Time=T_32
A_32 = 2*K3/(1+2*K3+K3^2);
%Third Impulse at Time=2*T_32
A_33 = (K3^2)/(1+2*K3+K3^2);
```

```
%Run DisPlantSecondOrderCLPDModel in Simulink
%This block diagram simulates a step input with
%input shaping for each individual mode
sim('DisPlantSecondOrderCLPDModel')
%Stead State value found from closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(t));
ss1=1.02*ones(size(t));
ss2=0.98*ones(size(t));
figure(1)
plot(t,yc/ss(temp(1,1)),'.',t,y1c/ss(temp(1,1)),':',...
t,y2c/ss(temp(1,1)),'-.',t,y3c/ss(temp(1,1)),'--',...
t,ycAllMode/ss(temp(1,1)),'-',t,ys,'k-.',...
t,ss1,'k',t,ss2,'k')
axis([0 2 -0.5 2.5])
title(...
'Simulation Reponse to Second Order(ZVD) Input Shaper')
xlabel('Time(seconds)')
ylabel('Normalized Angular Position of Theta 3')
%grid
legend('No Input Shaping','First Mode',...
   'Second Mode','Third Mode','All Three Modes')
figure(2)
plot(t,uc,'.',t,u1c,':',t,u2c,'-.',...
     t,u3c,'--',t,ucAllMode,'-')
axis([0 2 0 1.1])
title('System Input with Second Order(ZVD) Input Shaper')
```

```
xlabel('Time(seconds)')

ylabel('Input')

%grid

legend('No Input Shaping','First Mode',...

   'Second Mode','Third Mode','All Three Modes',0)

%Create the Trajectory file for ECP Hardware Experiments

%NOTE:  SET THE SEGMENT TIME EQUAL TO Ts ms !!!

MAG=input('Desired Number Counts: \n');

cd /, cd Research/ECPInputs

fid=fopen('DisPlantSecondOrderCLPD.trj','w');

length=size(ucAllMode);

fprintf(fid,'%f\r',[length(1,1);...

                    MAG*ucAllMode(1:1:length(1,1))]);

fclose(fid);

cd c:\Research\SingerShaper
```

Appendix C

MATLAB Codes(Modified Singer’s Method)

C.1 DisPlantFirstOrdModCLPD.m

```
%2nd Order Input Shaper for Closed Loop System
%With PD Control
%Singer’s Method
%DisPlantFirstOrdModCLPD.m

%Change to the Correct Directory
cd c:\Research\SingerShaper
%Clears All Old Variables
clear all, close all
%Clears Command Prompt
clc
%PD Control Gains
Kp=0.06;
Kd=0.75;
```

```
%Sampling Time of the ECP System
int=1; %Used To Specify Sampling
       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%1st Order Input Shaper for CL System for First Mode
%Damping Coefficient and System Natural Frequency
zeta1 = Z(1,1);
w1=Wn(1,1);
K1 = exp(-zeta1*pi/sqrt(1-zeta1^2));
T_12 = pi/(w1*sqrt(1-zeta1^2));deltaT1c=T_12;
%First Impulse at Time=0
A_11 = 1/(1+K1);A_11c=A_11;
%Second Impulse at Time=T_12
A_12 = K1/(1+K1);A_12c=A_12;
%1st Order Input Shaper for CL System for Second Mode
%Damping Coefficient and System Natural Frequency
zeta2 = Z(3,1);
w2=Wn(3,1);
K2 = exp(-zeta2*pi/sqrt(1-zeta2^2));
T_22 = pi/(w2*sqrt(1-zeta2^2));deltaT2c=T_22;
%First Impulse at Time=0
A_21 = 1/(1+K2);A_21c=A_21;
%Second Impulse at Time=T_22
A_22 = K2/(1+K2);A_22c=A_22;
%1st Order Input Shaper for CL System for Third Mode
```

```
%Damping Coefficient and System Natural Frequency
zeta3 = Z(5,1);
w3=Wn(5,1);
K3 = exp(-zeta3*pi/sqrt(1-zeta3^2));
T_32 - pi/(w3*sqrt(1-zeta3^2));deltaT3c=T_32;
%First Impulse at Time=0
A_31 = 1/(1+K3);A_31c=A_31;
%Second Impulse at Time=T_32
A_32 = K3/(1+K3);A_32c=A_32;
%Implmenting Singers Method in Discrete Time
%For First and Second Mode
%First Mode Freq and Damp Are Same
%Modify Second Mode Parameters
%So that T_12 is An Integer
%Multiple of mT_22(T for Shaper)
%Modified Parameters
mT_22=T_12/3;
mw2=Wn(3,1);
counter=0.0001;
temp1=(pi/(mw2*mT_22))^2;
while temp1 > 1

   temp1=(pi/((mw2+counter)*mT_22))^2;

   counter=counter+0.0001;

end
```

```
mw2=mw2+counter;

mzeta2=sqrt(1-(pi/(mw2*mT_22))^2);


mK2 = exp(-mzeta2*pi/sqrt(1-mzeta2^2));

mT_22 = pi/(mw2*sqrt(1-mzeta2^2));

%First Impulse at Time=0

mA_21 = 1/(1+mK2);

%Second Impulse at Time=T_22

mA_22 = mK2/(1+mK2);

%Implmenting Singers Method in Discrete Time

%For First and Second Mode

%First Mode Freq and Damp Are Same

%Modify Second Mode Parameters

%So that T_12 is An Integer

%Multiple of mT_23(T for Shaper)

%Modified Parameters

mT_32=T_12/4;

mw3=Wn(5,1);

counter=0.0001;

temp1=(pi/(mw3*mT_32))^2;

while temp1 > 1


   temp1=(pi/((mw3+counter)*mT_32))^2;


   counter=counter+0.0001;


end
```

```
mw3=mw3+counter;

mzeta3=sqrt(1-(pi/(mw3*mT_32))^2);


mK3 = exp(-mzeta3*pi/sqrt(1-mzeta3^2));

mT_32 = pi/(mw3*sqrt(1-mzeta3^2));

%First Impulse at Time=0

mA_31 = 1/(1+mK3);

%Second Impulse at Time=T_22

mA_32 = mK3/(1+mK3);

%Run DisPlantFirstOrdModCLPDModel in Simulink

%This block diagram simulates a step input

%with input shaping for each individual mode

sim('DisPlantFirstOrdModCLPDModel')

%Simulate Unmodified to Compare Results

sim('DisPlantFirstOrderCLPDModel')

%Stead State value found for closed loop system

[ss,time]=step(sysdtcl);temp=size(ss);

ys=ones(size(t));

ss1=1.02*ones(size(t));

ss2=0.98*ones(size(t));

figure(1)

plot(t,yc/ss(temp(1,1)),'-',t,y1c/ss(temp(1,1)),':',...

t,y2c/ss(temp(1,1)),'-.',t,y3c/ss(temp(1,1)),'--',...

t,ycAllMode/ss(temp(1,1)),'k',t,ys,'k-.',...

t,ss1,'k',t,ss2,'k')

title('Closed Loop System Reponse to Step Input')

xlabel('Time(seconds)')
```

```
ylabel('Normalized Angular Position of Theta 3')
grid
legend('No Input Shaping','First Mode',...
   'Second Mode','Third Mode','All Three Modes')
figure(2)
plot(t,uc,'-',t,u1c,':',t,u2c,'-.',t,u3c,'--',...
t,ucAllMode,'k')
axis([-1 2 0 1.1])
title('System Input')
xlabel('Time(seconds)')
ylabel('Input')
grid
legend('No Input Shaping','First Mode',...
   'Second Mode','Third Mode','All Three Modes')
figure(3)
plot(t,yc/ss(temp(1,1)),'.',ts,y12/ss(temp(1,1)),':',...
ts,y13/ss(temp(1,1)),'-.',t,ycAllMode/ss(temp(1,1)),...
'-',t,ys,'k-.',t,ss1,'k',t,ss2,'k')
axis([0 1 -0.5 2.5])
title(...
'Reponse to First Order(ZV) and Modified ZV Input Shaper')
xlabel('Time(seconds)')
ylabel('Normalized Angular Position of Theta 3')
%grid
legend('No Input Shaping',...
'First/Second Modified Discrete',...
'First/Third Modified Discrete',...
'All Three Modes Continuous')
figure(4)
plot(t,uc,'.',t22,u12,':',t32,u13,'-.',t,ucAllMode,'-')
axis([0 1 0 1.1])
title(...
'System Input with Modified First Order(ZV) Input Shaper')
xlabel('Time(seconds)')
ylabel('Input')
%grid
legend('No Input Shaping',...
'First/Second Modified Discrete',...
'First/Third Modified Discrete',...
'All Three Modes Continuous',0)
%Create the Trajectory file for ECP Hardware Experiments
%NOTE: SET THE SEGMENT TIME EQUAL TO mT_22 or mT_23 mS !!
MAG=input('Desired Number Counts: \n');
cd /, cd Research/ECPInputs
fid=fopen('DisPlantFirstOrdModCLPD12.trj','w');
length=size(u12);
fprintf(fid,'%f\r',[length(1,1);MAG*u12(1:1:length(1,1))]);
fclose(fid);
cd c:\Research\SingerShaper
cd /, cd Research/ECPInputs
fid=fopen('DisPlantFirstOrdModCLPD13.trj','w');
length=size(u13);
fprintf(fid,'%f\r',[length(1,1);MAG*u13(1:1:length(1,1))]);
fclose(fid);
```

```
cd c:\Research\SingerShaper
```

C.2 DisPlantSecOrdModCLPD.m

```
%2nd Order Input Shaper for Closed Loop System
%With PD Control
%DisPlantSecOrdModCLPD.m

%Change to the Correct Directory
cd c:\Research\SingerShaper
%Clears All Old Variables
clear all, close all
%Clears Command Prompt
clc
%PD Control Gains
Kp=0.06;
Kd=0.75;
%Sampling Time of the ECP System
int=1; %Used To Specify Sampling
       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%1st Order Input Shaper for CL System for First Mode
%Damping Coefficient and System Natural Frequency
zeta1 = Z(1,1);
w1=Wn(1,1);
```

```
K1 = exp(-zeta1*pi/sqrt(1-zeta1^2));
T_12 = pi/(w1*sqrt(1-zeta1^2));
T_13 = 2*T_12;
%First Impulse at Time=0
A_11 = 1/(1+2*K1+K1^2);
%Second Impulse at Time=T_12
A_12 = 2*K1/(1+2*K1+K1^2);
%Third Impulse at Time=2*T_12
A_13 = (K1^2)/(1+2*K1+K1^2);
%1st Order Input Shaper for CL System for Second Mode
%Damping Coefficient and System Natural Frequency
zeta2 = Z(3,1);
w2=Wn(3,1);
K2 = exp(-zeta2*pi/sqrt(1-zeta2^2));
T_22 = pi/(w2*sqrt(1-zeta2^2));
T_23 = 2*T_22;
%First Impulse at Time=0
A_21 = 1/(1+2*K2+K2^2);
%Second Impulse at Time=T_22
A_22 = 2*K2/(1+2*K2+K2^2);
%Third Impulse at Time=2*T_22
A_23 = (K2^2)/(1+2*K2+K2^2);
%1st Order Input Shaper for CL System for Third Mode
%Damping Coefficient and System Natural Frequency
zeta3 = Z(5,1);
w3=Wn(5,1);
K3 = exp(-zeta3*pi/sqrt(1-zeta3^2));
```

```
T_32 = pi/(w3*sqrt(1-zeta3^2));
T_33 = 2*T_32;
%First Impulse at Time=0
A_31 = 1/(1+2*K3+K3^2);
%Second Impulse at Time=T_32
A_32 = 2*K3/(1+2*K3+K3^2);
%Third Impulse at Time=2*T_32
A_33 = (K3^2)/(1+2*K3+K3^2);
%Implmenting Singers Method in Discrete Time
%For First and Second Mode
%First Mode Freq and Damp Are Same
%Modify Second Mode Parameters
%So that T_12 is An Integer
%Multiple of mT_22(T for Shaper)
%Modified Parameters
mT_22=T_12/3;
mw2=Wn(3,1);
counter=0.0001;
temp1=(pi/(mw2*mT_22))^2;
while temp1 > 1

   temp1=(pi/((mw2+counter)*mT_22))^2;

   counter=counter+0.0001;

end
mw2=mw2+counter;
```

```
mzeta2=sqrt(1-(pi/(mw2*mT_22))^2);


mK2 = exp(-mzeta2*pi/sqrt(1-mzeta2^2));

mT_22 = pi/(mw2*sqrt(1-mzeta2^2));

mT_23 = 2*mT_22;

%First Impulse at Time=0

mA_21 = 1/(1+2*mK2+mK2^2);

%Second Impulse at Time=T_22

mA_22 = 2*mK2/(1+2*mK2+mK2^2);

%Third Impulse at Time=2*T_22

mA_23 = (mK2^2)/(1+2*mK2+mK2^2);

%Implmenting Singers Method in Discrete Time

%For First and Second Mode

%First Mode Freq and Damp Are Same

%Modify Second Mode Parameters

%So that T_12 is An Integer

%Multiple of mT_23(T' for Shaper)

%Modified Parameters

mT_32=T_12/4;

mw3=Wn(5,1);

counter=0.0001;

temp1=(pi/(mw3*mT_32))^2;

while temp1 > 1


   temp1=(pi/((mw3+counter)*mT_32))^2;


   counter=counter+0.0001;
```

```
end
mw3=mw3+counter;
mzeta3=sqrt(1-(pi/(mw3*mT_32))^2);

mK3 = exp(-mzeta3*pi/sqrt(1-mzeta3^2));
mT_32 = pi/(mw3*sqrt(1-mzeta3^2));
mT_33 = 2*mT_32;
%First Impulse at Time=0
mA_31 = 1/(1+2*mK3+mK3^2);
%Second Impulse at Time=T_22
mA_32 = 2*mK3/(1+2*mK3+mK3^2);
%Third Impulse at Time=2*T_22
mA_33 = (mK3^2)/(1+2*mK3+mK3^2);
%Run DisPlantSecOrdModCLPDModel in Simulink
%This block diagram simulates a step input
%with input shaping for each individual mode
sim('DisPlantSecOrdModCLPDModel')
%Simulate Unmodified to Compare Results
sim('DisPlantSecondOrderCLPDModel')
%Stead State value found for closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(t));
ss1=1.02*ones(size(t));
ss2=0.98*ones(size(t));
figure(1)
plot(t,yc/ss(temp(1,1)),'-',t,y1c/ss(temp(1,1)),':',...
t,y2c/ss(temp(1,1)),'-.',t,y3c/ss(temp(1,1)),'--',...
t,ycAllMode/ss(temp(1,1)),'k',t,ys,'k-.',t,ss1,'k',...
t,ss2,'k')
title('Closed Loop System Reponse to Step Input')
xlabel('Time(seconds)')
ylabel('Normalized Angular Position of Theta 3')
grid
legend('No Input Shaping','First Mode',...
   'Second Mode','Third Mode','All Three Modes')
figure(2)
plot(t,uc,'-',t,u1c,':',t,u2c,'-.',t,u3c,'--',...
t,ucAllMode,'k')
axis([-1 2 0 1.1])
title('System Input')
xlabel('Time(seconds)')
ylabel('Input')
grid
legend('No Input Shaping','First Mode',...
   'Second Mode','Third Mode','All Three Modes')
figure(3)
plot(t,yc/ss(temp(1,1)),'.',ts,y12/ss(temp(1,1)),':',...
ts,y13/ss(temp(1,1)),'-.',t,ycAllMode/ss(temp(1,1)),...
'-',t,ys,'k-.',t,ss1,'k',t,ss2,'k')
axis([0 2 -0.5 2.5])
title(...
'Reponse to Second Order(ZVD) and Modified ZVD Shaper')
xlabel('Time(seconds)')
```

```
ylabel('Normalized Angular Position of Theta 3')
%grid
legend('No Input Shaping',...
'First/Second Modified Discrete',...
'First/Third Modified Discrete',...
'All Three Modes Continuous')
figure(4)
plot(t,uc,'.',t22,u12,':',t32,u13,'-.',t,ucAllMode,'-')
axis([0 2 0 1.1])
title(...
'System Input with Modified Second Order(ZVD) Input Shaper')
xlabel('Time(seconds)')
ylabel('Input')
%grid
legend('No Input Shaping',...
'First/Second Modified Discrete',...
'First/Third Modified Discrete',...
'All Three Modes Continuous',0)
%Create the Trajectory file for ECP Hardware Experiments
%NOTE:  SET THE SEGMENT TIME EQUAL TO TsmS !!!
MAG=input('Desired Number Counts: \n');
cd /, cd Research/ECPInputs
fid=fopen('DisPlantSecOrdModCLPD12.trj','w');
length=size(u12);
fprintf(fid,'%f\r',[length(1,1);MAG*u12(1:1:length(1,1))]);
fclose(fid);
cd c:\Research\SingerShaper
cd /, cd Research/ECPInputs

fid=fopen('DisPlantSecOrdModCLPD13.trj','w');

length=size(u13);

fprintf(fid,'%f\r',[length(1,1);MAG*u13(1:1:length(1,1))]);

fclose(fid);

cd c:\Research\SingerShaper
```

Appendix D

MATLAB Codes(Tuttle's Method)

D.1 DisPlantFirstOrderDisCLPD.m

```
%1nd Order Input Shaper for Closed Loop System
%With PD Control
%DisPlantFirstOrderDisCLPD.m

%Change to the Correct Directory
cd c:\Research\TuttleShaper
%Clears All Old Variables
clear all, close all
%Clears Command Prompt
clc
%PD Control Gains
Kp=0.06;
Kd=0.75;
%Sampling Time of the ECP System
int=1; %Used To Specify Sampling
       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%Amplitude of the Step Input
AMP=1;
%Closed Loop Natural Frequencies of each mode
W1=Wn(1,1);
W2=Wn(3,1);
W3=Wn(5,1);
%Closed Loop Modal Damping Ratios of each mode
D1=Z(1,1);
D2=Z(3,1);
D3=Z(5,1);
%Closed Loop Damped Natural Frequency
Wd1=W1*sqrt(1-D1^2);
Wd2=W2*sqrt(1-D2^2);
Wd3=W3*sqrt(1-D3^2);
hold on
%Discrete Sampling Period
for T=0:0.0005:0.2;
%Input Shaper Zeros
p1=exp(-D1.*W1.*T).*exp(Wd1.*T.*j);
p1s=exp(-D1.*W1.*T).*exp(-Wd1.*T.*j);
p2=exp(-D2.*W2.*T).*exp(Wd2.*T.*j);
p2s=exp(-D2.*W2.*T).*exp(-Wd2.*T.*j);
p3=exp(-D3.*W3.*T).*exp(Wd3.*T.*j);
p3s=exp(-D3.*W3.*T).*exp(-Wd3.*T.*j);
```

```
%Impulse amplitudes
Aa=conv([1  -p1],conv([1  -p1s],conv([1  -p2],...
   conv([1  -p2s],conv([1  -p3],[1  -p3s])))));
Aa=real(Aa);
figure(1),
plot(T,Aa(1),’.’,T,Aa(2),’.’,T,Aa(3),’.’,T,Aa(4),’.’,...
T,Aa(5),’.’,T,Aa(6),’.’,T,Aa(7),’.’)
end
title(...
’Impulse Amplitudes for First Order Zero Placement’)
xlabel(’Impulse Spacing, T(seconds)’)
ylabel(’Impulse Amplitudes’)
grid
hold off,pause(1)
%Find the Impulse Amplitudes for a T that yields
%all Positive Impulse Amplitudes
T=0.125;
%Input Shaper Zeros
p1=exp(-D1*W1*T)*exp(Wd1*T*j);
p1s=exp(-D1*W1*T)*exp(-Wd1*T*j);
p2=exp(-D2*W2*T)*exp(Wd2*T*j);
p2s=exp(-D2*W2*T)*exp(-Wd2*T*j);
p3=exp(-D3*W3*T)*exp(Wd3*T*j);
p3s=exp(-D3*W3*T)*exp(-Wd3*T*j);
%Impulse amplitudes
Aa=conv([1  -p1],conv([1  -p1s],conv([1  -p2],...
   conv([1  -p2s],conv([1  -p3],[1  -p3s])))));
```

```
Aa=real(Aa);
%Scaling Constant CC
CC=(Aa(1)+Aa(2)+Aa(3)+Aa(4)+Aa(5)+Aa(6)+Aa(7))^(-1);
CC=real(CC);
%Run DisPlantFirstOrderDisCLPDModel in Simulink
%This block diagram simulates a step input with
%input shaping for each individual mode
sim('DisPlantFirstOrderDisCLPDModel')
%Stead State value found for closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(dTs));
ss1=ys*1.02;
ss2=ys*0.98;
figure(2)
plot(dTs,ystep/ss(temp(1,1)),dTs,ys,'k-.',...
   dTs,ss1,'k',dTs,ss2,'k')
axis([0 1 0 1.2])
title(...
'Simulation Reponse to First Order Zero Placement Shaper')
xlabel('Time(seconds)')
ylabel('Normalized Angular Position of Theta 3')
%grid
figure(3)
plot(dT,u,'-')
axis([0 1 0 1.1])
title(...
'System Input with First Order Zero Placement Shaper')
```

```
xlabel('Time(seconds)')
ylabel('Input')
%grid
%Create the Trajectory file for ECP Hardware Experiments
%NOTE:  SET THE SEGMENT TIME EQUAL TO T mS !
MAG=input('Desired Number Counts: \n');
cd /, cd Research/ECPInputs
fid=fopen('DisPlantFirstOrderDisCLPD.trj','w');
length=size(u);
fprintf(fid,'%f\r',[length(1,1);MAG*u(1:1:length(1,1))]);
fclose(fid);
cd c:\Research\TuttleShaper
```

D.2 DisPlantSecondOrderDisCLPD.m

```
%1nd Order Input Shaper for Closed Loop System
%With PD Control
%DisPlantSecondOrderDisCLPD.m

%Change to the Correct Directory
cd c:\Research\TuttleShaper
%Clears All Old Variables
clear all, close all
%Clears Command Prompt
clc
%PD Control Gains
Kp=0.06;
```

```
Kd=0.75;
%Sampling Time of the ECP System
int=1; %Used To Specify Sampling
       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%Amplitude of the Step Input
AMP=1;
%Closed Loop Natural Frequencies of each mode
W1=Wn(1,1);
W2=Wn(3,1);
W3=Wn(5,1);
%Closed Loop Modal Damping Ratios of each mode
D1=Z(1,1);
D2=Z(3,1);
D3=Z(5,1);
%Closed Loop Damped Natural Frequency
Wd1=W1*sqrt(1-D1^2);
Wd2=W2*sqrt(1-D2^2);
Wd3=W3*sqrt(1-D3^2);
hold on
%Discrete Sampling Period
for T=0:0.0005:0.2;
%Input Shaper Zeros
p1=exp(-D1.*W1.*T).*exp(Wd1.*T.*j);
p1s=exp(-D1.*W1.*T).*exp(-Wd1.*T.*j);
```

```
p2=exp(-D2.*W2.*T).*exp(Wd2.*T.*j);
p2s=exp(-D2.*W2.*T).*exp(-Wd2.*T.*j);
p3=exp(-D3.*W3.*T).*exp(Wd3.*T.*j);
p3s=exp(-D3.*W3.*T).*exp(-Wd3.*T.*j);
%Impulse amplitudes
Aa=conv([1  -p1],conv([1  -p1],conv([1  -p1s],...
conv([1  -p1s],conv([1  -p2],conv([1  -p2],...
conv([1  -p2s],conv([1  -p2s],conv([1  -p3],...
conv([1  -p3],conv([1  -p3s],[1  -p3s]))))))))))));
Aa=real(Aa);
figure(1),
plot(T,Aa(1),’.’,T,Aa(2),’.’,T,Aa(3),’.’,T,Aa(4),’.’,...
T,Aa(5),’.’,T,Aa(6),’.’,T,Aa(7),’.’,T,Aa(8),’.’,...
T,Aa(9),’.’,T,Aa(10),’.’,T,Aa(11),’.’,T,Aa(12),’.’,...
T,Aa(13),’.’)
end
title(...
’Impulse Amplitudes for Second Order Zero Placement’)
xlabel(’Impulse Spacing, T(seconds)’)
ylabel(’Impulse Amplitudes’)
grid
hold off,pause(1)
%Find the Impulse Amplitudes for a T that yields
%all Positive Impulse Amplitudes
T=0.125;
%Input Shaper Zeros
p1=exp(-D1*W1*T)*exp(Wd1*T*j);
```

```
p1s=exp(-D1*W1*T)*exp(-Wd1*T*j);
p2=exp(-D2*W2*T)*exp(Wd2*T*j);
p2s=exp(-D2*W2*T)*exp(-Wd2*T*j);
p3=exp(-D3*W3*T)*exp(Wd3*T*j);
p3s=exp(-D3*W3*T)*exp(-Wd3*T*j);
%Impulse amplitudes
Aa=conv([1  -p1],conv([1  -p1],conv([1  -p1s],...
   conv([1  -p1s],conv([1  -p2],conv([1  -p2],...
   conv([1  -p2s],conv([1  -p2s],conv([1  -p3],...
   conv([1  -p3],conv([1  -p3s],...
   [1  -p3s])))))))))));
Aa=real(Aa);
%Scaling Constant CC
CC=(Aa(1)+Aa(2)+Aa(3)+Aa(4)+Aa(5)+Aa(6)+Aa(7)+Aa(8)...
    +Aa(9)+Aa(10)+Aa(11)+Aa(12)+Aa(13))^(-1);
CC=real(CC);
%Run DisPlantSecondOrderDisCLPDModel in Simulink
%This block diagram simulates a step input with input
%shaping for each individual mode
sim('DisPlantSecondOrderDisCLPDModel')
%Stead State value found for closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(dTs));
ss1=ys*1.02;
ss2=ys*0.98;
figure(2)
plot(dTs,ystep/ss(temp(1,1)),dTs,ys,'k-.',...
   dTs,ss1,'k',dTs,ss2,'k')
axis([0 2 0 1.2])
title(...
'Simulation Reponse to Second Order Zero Placement')
xlabel('Time(seconds)')
ylabel('Normalized Angular Position of Theta 3')
%grid
figure(3)
plot(dT,u,'-')
axis([0 2 0 1.1])
title(...
'System Input with Second Order Zero Placement')
xlabel('Time(seconds)')
ylabel('Input')
%Create the Trajectory file ECP Hardware Experiments
%NOTE:  SET THE SEGMENT TIME EQUAL TO T mS !!!!
MAG=input('Desired Number Counts: \n');
cd /, cd Research/ECPInputs
fid=fopen('DisPlantSecondOrderDisCLPD.trj','w');
length=size(u);
fprintf(fid,'%f\r',[length(1,1);MAG*u(1:1:length(1,1))]);
fclose(fid);
cd c:\Research\TuttleShaper
```

D.3 DisPlantSecOrderMode1DisCLPD.m

```
%1nd Order Input Shaper for Closed Loop System
```

```
%With PD Control
%DisPlantSecOrderModelDisCLPD.m

%Change to the Correct Directory
cd c:\Research\TuttleShaper
%Clears All Old Variables
clear all, close all
%Clears Command Prompt
clc
%PD Control Gains
Kp=0.06;
Kd=0.75;
%Sampling Time of the ECP System
int=1; %Used To Specify Sampling
       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%Amplitude of the Step Input
AMP=1;
%Closed Loop Natural Frequencies of each mode
W1=Wn(1,1);
W2=Wn(3,1);
W3=Wn(5,1);
%Closed Loop Modal Damping Ratios of each mode
D1=Z(1,1);
D2=Z(3,1);
```

```
D3=Z(5,1);
%Closed Loop Damped Natural Frequency
Wd1=W1*sqrt(1-D1^2);
Wd2=W2*sqrt(1-D2^2);
Wd3=W3*sqrt(1-D3^2);
hold on
%Discrete Sampling Period
for T=0:0.0005:0.2;
%Input Shaper Zeros
p1=exp(-D1.*W1.*T).*exp(Wd1.*T.*j);
p1s=exp(-D1.*W1.*T).*exp(-Wd1.*T.*j);
p2=exp(-D2.*W2.*T).*exp(Wd2.*T.*j);
p2s=exp(-D2.*W2.*T).*exp(-Wd2.*T.*j);
p3=exp(-D3.*W3.*T).*exp(Wd3.*T.*j);
p3s=exp(-D3.*W3.*T).*exp(-Wd3.*T.*j);
%Impulse amplitudes
Aa=conv([1  -p1],conv([1  -p1],conv([1  -p1s],...
   conv([1  -p1s],conv([1  -p2],conv([1  -p2s],...
   conv([1  -p3],[1  -p3s])))))));
Aa=real(Aa);
figure(1),
plot(T,Aa(1),’.’,T,Aa(2),’.’,T,Aa(3),’.’,T,Aa(4),’.’,...
T,Aa(5),’.’,T,Aa(6),’.’,T,Aa(7),’.’,T,Aa(8),’.’,...
T,Aa(9),’.’)
end
title(...
’Impulse Amplitudes for 1st/2nd Order Zero Placement’)
```

```
xlabel('Impulse Spacing, T(seconds)')
ylabel('Impulse Amplitudes')
grid
hold off,pause(1)
%Find the Impulse Amplitudes for a T that yields
%all Positive Impulse Amplitudes
T=0.140;
%Input Shaper Zeros
p1=exp(-D1*W1*T)*exp(Wd1*T*j);
p1s=exp(-D1*W1*T)*exp(-Wd1*T*j);
p2=exp(-D2*W2*T)*exp(Wd2*T*j);
p2s=exp(-D2*W2*T)*exp(-Wd2*T*j);
p3=exp(-D3*W3*T)*exp(Wd3*T*j);
p3s=exp(-D3*W3*T)*exp(-Wd3*T*j);
%Impulse amplitudes
Aa=conv([1  -p1],conv([1  -p1],conv([1  -p1s],...
   conv([1  -p1s],conv([1  -p2],conv([1  -p2s],...
   conv([1  -p3],[1  -p3s])))))));
Aa=real(Aa);
%Scaling Constant CC
CC=(Aa(1)+Aa(2)+Aa(3)+Aa(4)+Aa(5)+Aa(6)+Aa(7)...
   +Aa(8)+Aa(9))^(-1);
CC=real(CC);
%Run DisPlantSecOrdModelDisCLPDModel in Simulink
%This block diagram simulates a step input with
%input shaping for each individual mode
sim('DisPlantSecOrdModelDisCLPDModel')
```

```
%Stead State value found for closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(dTs));
ss1=ys*1.02;
ss2=ys*0.98;
figure(2)
plot(dTs,ystep/ss(temp(1,1)),dTs,ys,'k-.',...
   dTs,ss1,'k',dTs,ss2,'k')
axis([0 2 0 1.2])
title(...
'Simulation Reponse to 1st/2nd Order Zero Placement')
xlabel('Time(seconds)')
ylabel('Normalized Angular Position of Theta 3')
%grid
figure(3)
plot(dT,u,'-')
axis([0 2 0 1.1])
title(...
'System Input with 1st/2nd Order Zero Placement')
xlabel('Time(seconds)')
ylabel('Input')
%Create the Trajectory file ECP Hardware Experiments
%NOTE:  SET THE SEGMENT TIME EQUAL TO T mS !!!!
MAG=input('Desired Number Counts: \n');
cd /, cd Research/ECPInputs
fid=fopen('DisPlantSecOrderModelDisCLPD.trj','w');
length=size(u);
```

```
fprintf(fid,'%f\r',[length(1,1);MAG*u(1:1:length(1,1))]);

fclose(fid);

cd c:\Research\TuttleShaper
```

Appendix E

MATLAB Codes(Singhose's Method)

E.1 DisPlantEIOneHumpShaper.m

```
%EI Input Shaper for Closed Loop System

%With PD Control

%DisPlantEIOneHumpShaper.m


%Change to the Correct Directory

cd c:\Research\SinghoseShaper

%Clears All Old Variables

clear all,close all

%Clears Command Prompt

clc

%PD Control Gains

Kp=0.06;

Kd=0.75;

%Sampling Time of the ECP System

int=1; %Used To Specify Sampling

       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%Set the Maximum Level of Vibration
V=0.05;
%Natural Frequency and Damping Ratio of the 1st Mode
z1=Z(1,1);
w1 = Wn(1,1);
%Damped Period of Vibration
Td1=2*pi/(w1*sqrt(1-z1^2));
%First Impulse
A_11=0.2497+0.2496*V+0.8001*z1+...
     1.233*V*z1+0.496*(z1^2)+3.173*V*(z1^2);
T_11=0;
%Third Impulse
A_13=0.2515+0.2147*V-0.8325*z1+...
     1.415*V*z1+0.8518*(z1^2)-4.901*V*(z1^2);
T_13=Td1;
%Second Impulse
A_12=1-(A_11+A_13);
T_12=(0.5+0.4616*V*z1+4.262*V*(z1^2)+1.756*V*(z1^3)+ ...
8.578*(V^2)*z1-108.6*(V^2)*(z1^2)+337*(V^2)*(z1^3))*Td1;
%Natural Frequency and Damping Ratio of the 2nd Mode
z2=Z(3,1);
w2 = Wn(3,1);
%Damped Period of Vibration
Td2=2*pi/(w2*sqrt(1-z2^2));
```

```
%First Impulse

A_21=0.2497+0.2496*V+0.8001*z2+...

     1.233*V*z2+0.496*(z2^2)+3.173*V*(z2^2);

T_21=0;

%Third Impulse

A_23=0.2515+0.2147*V-0.8325*z2+...

     1.415*V*z2+0.8518*(z2^2)-4.901*V*(z2^2);

T_23=Td2;

%Second Impulse

A_22=1-(A_21+A_23);

T_22=(0.5+0.4616*V*z2+4.262*V*(z2^2)+1.756*V*(z2^3)+ ...

8.578*(V^2)*z2-108.6*(V^2)*(z2^2)+337*(V^2)*(z2^3))*Td2;

%Natural Frequency and Damping Ratio of the 3rd Mode

z3=Z(5,1);

w3 = Wn(5,1);

%Damped Period of Vibration

Td3=2*pi/(w3*sqrt(1-z3^2));

%First Impulse

A_31=0.2497+0.2496*V+0.8001*z3+...

     1.233*V*z3+0.496*(z3^2)+3.173*V*(z3^2);

T_31=0;

%Third Impulse

A_33=0.2515+0.2147*V-0.8325*z3+...

     1.415*V*z3+0.8518*(z3^2)-4.901*V*(z3^2);

T_33=Td3;

%Second Impulse

A_32=1-(A_31+A_33);
```

```
T_32=(0.5+0.4616*V*z3+4.262*V*(z3^2)+1.756*V*(z3^3)+ ...
8.578*(V^2)*z3-108.6*(V^2)*(z3^2)+337*(V^2)*(z3^3))*Td3;
%Simulate the System Response to a Unit Step Input
%with and without Input Shaping
sim('DisPlantEIOneHumpShaperModel')
%Stead State value found from closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(t));
ss1=1.02*ones(size(t));
ss2=0.98*ones(size(t));
figure(1)
plot(t,y/ss(temp(1,1)),'.',t,y1/ss(temp(1,1)),':',...
t,y2/ss(temp(1,1)),'-.',t,y3/ss(temp(1,1)),'--',...
t,yAllMode/ss(temp(1,1)),'-',t,ys,'k-.',...
t,ss1,'k',t,ss2,'k')
axis([0 2 -0.5 2.5])
title(...
'Simulation Response with a One Hump EI Input Shaper')
xlabel('Time(seconds)'),
ylabel('Normalized Angular Position of Theta 3')
legend('No Input Shaper','1st Mode'...
   ,'2nd Mode','3rd Mode','All Modes')
figure(2)
plot(t,u,'.',t,u1,':',t,u2,'-.',t,u3,'--',t,uAllMode,'-')
axis([0 2 0 1.1])
title(...
'System Input with and without a One Hump EI Input Shaper')
```

```
xlabel('Time(seconds)'),
ylabel('Input')
legend('No Input Shaper','1st Mode'...
   ,'2nd Mode','3rd Mode','All Modes',0)
%Create the Trajectory file for ECP Hardware Experiments
%NOTE:  SET THE SEGMENT TIME EQUAL TO Ts mS !
MAG=input('Desired Number Counts: \n');
cd /, cd Research/ECPInputs
fid=fopen('DisPlantEIOneHumpShaper.trj','w');
length=size(uAllMode);
fprintf(fid,'%f\r',[length(1,1);...
                    MAG*uAllMode(1:1:length(1,1))]);
fclose(fid);
cd c:\Research\SinghoseShaper
```

E.2 DisPlantUMZVShaper.m

```
%UM-ZV Input Shaper for Closed Loop System
%With PD Control
%DisPlantUMZVShaper.m

%Change to the Correct Directory
cd c:\Research\SinghoseShaper
%Clears All Old Variables
clear all,close all
%Clears Command Prompt
clc
```

```
%PD Control Gains
Kp=0.06;
Kd=0.75;
%Sampling Time of the ECP System
int=1; %Used To Specify Sampling
    %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
%Natural Frequency and Damping Ratio of the 1st Mode
z1=Z(1,1);
w1 = Wn(1,1);
%Period of Vibration
T1=2*pi/w1;
%First Impulse
A_11=1;
T_11=0;
%Second Impulse
A_12=-1;
T_12=(0.16724+0.27242*z1+0.20345*z1^2)*T1;
%Third Impulse
A_13=1;
T_13=(0.33323+0.00533*z1+0.17914*z1^2+0.20125*z1^3)*T1;
%Natural Frequency and Damping Ratio of the 2nd Mode
z2=Z(3,1);
w2 = Wn(3,1);
%Period of Vibration
```

```
T2=2*pi/w2;
%First Impulse
A_21=1;
T_21=0;
%Second Impulse
A_22=-1;
T_22=(0.16724+0.27242*z2+0.20345*z2^2)*T2;
%Third Impulse
A_23=1;
T_23=(0.33323+0.00533*z2+0.17914*z2^2+0.20125*z2^3)*T2;
%Natural Frequency and Damping Ratio of the 3rd Mode
z3=Z(5,1);
w3 = Wn(5,1);
%Period of Vibration
T3=2*pi/w3;
%First Impulse
A_31=1;
T_31=0;
%Second Impulse
A_32=-1;
T_32=(0.16724+0.27242*z3+0.20345*z3^2)*T3;
%Third Impulse
A_33=1;
T_33=(0.33323+0.00533*z3+0.17914*z3^2+0.20125*z3^3)*T3;
%Simulate the System Response to a Unit Step Input
%with and without Input Shaping
sim('DisPlantUMZVShaperModel')
```

```
%Stead State value found from closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(t));
ss1=1.02*ones(size(t));
ss2=0.98*ones(size(t));
figure(1)
plot(t,y/ss(temp(1,1)),'.',t,y1/ss(temp(1,1)),':',...
t,y2/ss(temp(1,1)),'-.',t,y3/ss(temp(1,1)),'--',...
t,yAllMode/ss(temp(1,1)),'-',t,ys,'k-.',...
t,ss1,'k',t,ss2,'k')
axis([0 1 -0.5 2.5])
title('Step Response with a Unity Magnitude Input Shaper')
xlabel('Time(seconds)'),
ylabel('Normalized Angular Position of Theta 3')
legend('No Input Shaper','1st Mode'...
   ,'2nd Mode','3rd Mode','All Modes')
figure(2)
plot(t,u,'.',t,u1,':',t,u2,'-.',t,u3,'--',t,uAllMode,'-')
axis([0 0.6 -1 2])
title(...
'System Input with and without Unity Magnitude Shaper')
xlabel('Time(seconds)'),ylabel('Input')
legend('No Input Shaper','1st Mode',...
   '2nd Mode','3rd Mode','All Modes',0)
%Create the Trajectory file the ECP Hardware Experiments
%NOTE:  SET THE SEGMENT TIME EQUAL TO 1mS !
MAG=input('Desired Number Counts: \n');
```

```
cd /, cd Research/ECPInputs
fid=fopen('DisPlantUMZVShaper.trj','w');
length=size(u1);
fprintf(fid,'%f\r',[length(1,1);MAG*u1(1:1:length(1,1))]);
fclose(fid);
```

E.3 DisPlantPSZVShaper.m

```
%PS-ZV Input Shaper for Closed Loop System
%With PD Control
%DisPlantPSZVShaper.m

%Change to the Correct Directory
cd c:\Research\SinghoseShaper
%Clears All Old Variables
clear all,close all
%Clears Command Prompt
clc
%PD Control Gains
Kp=0.06;
Kd=0.75;
%Sampling Time of the ECP System
int=1; %Used To Specify Sampling
       %in Integer Multiples of
       %the minimum Ts
%Build Discrete Model
Discrete
```

```
%%Natural Frequency and Damping Ratio of the 1st Mode
z1=Z(1,1);
w1 = Wn(1,1);
%Period of Vibration
T1=2*pi/w1;
%First Impulse
A_11=1;
T_11=0;
%Second Impulse
A_12=-2;
T_12=(0.2097+0.22441*z1+0.08028*z1^2+0.23124*z1^3)*T1;
%Third Impulse
A_13=2;
T_13=(0.29013+0.09557*z1+0.10346*z1^2+0.24624*z1^3)*T1;
%%Natural Frequency and Damping Ratio of the 2nd Mode
z2=Z(3,1);
w2 = Wn(3,1);
%Period of Vibration
T2=2*pi/w2;
%First Impulse
A_21=1;
T_21=0;
%Second Impulse
A_22=-2;
T_22=(0.2097+0.22441*z2+0.08028*z2^2+0.23124*z2^3)*T2;
%Third Impulse
A_23=2;
```

```
T_23=(0.29013+0.09557*z2+0.10346*z2^2+0.24624*z2^3)*T2;
%%Natural Frequency and Damping Ratio of the 3rd Mode
z3=Z(5,1);
w3 = Wn(5,1);
%Period of Vibration
T3=2*pi/w3;
%First Impulse
A_31=1;
T_31=0;
%Second Impulse
A_32=-2;
T_32=(0.2097+0.22441*z3+0.08028*z3^2+0.23124*z3^3)*T3;
%Third Impulse
A_33=2;
T_33=(0.29013+0.09557*z3+0.10346*z3^2+0.24624*z3^3)*T3;
%Simulate the System Response to a Unit Step Input
%with and without Input Shaping
sim('DisPlantPSZVShaperModel')
%Stead State value found from closed loop system
[ss,time]=step(sysdtcl);temp=size(ss);
ys=ones(size(t));
ss1=1.02*ones(size(t));
ss2=0.98*ones(size(t));
figure(1)
plot(t,y/ss(temp(1,1)),'.',t,y1/ss(temp(1,1)),':',...
t,y2/ss(temp(1,1)),'-.',t,y3/ss(temp(1,1)),'--',...
t,yAllMode/ss(temp(1,1)),'-',t,ys,'k-.',...
t,ss1,'k',t,ss2,'k')
axis([0 1 -0.5 2.5])
title('Step Response with a Partial Sum Input Shaper')
xlabel('Time(seconds)'),
ylabel('Normalized Angular Position of Theta 3')
legend('No Input Shaper','1st Mode'...
   ,'2nd Mode','3rd Mode','All Modes')
figure(2)
plot(t,u,'.',t,u1,':',t,u2,'-.',t,u3,'--',t,uAllMode,'-')
axis([0 0.5 -2 2])
title(...
'System Input with and without a Partial Sum Shaper')
xlabel('Time(seconds)'),ylabel('Input')
legend('No Input Shaper','1st Mode'...
   ,'2nd Mode','3rd Mode','All Modes',0)
%Create the Trajectory file for ECP Hardware Experiments
%NOTE:  SET THE SEGMENT TIME EQUAL TO Ts mS !
MAG=input('Desired Number Counts: \n');
cd /, cd Research/ECPInputs
fid=fopen('DisPlantPSZVShaper.trj','w');
length=size(uAllMode);
fprintf(fid,'%f\r',[length(1,1);...
                    MAG*uAllMode(1:1:length(1,1))]);
fclose(fid);
cd c:\Research\SinghoseShaper
```

VITA

Ivan Leonel Blanco

Candidate for the Degree of

Master of Science

Thesis: RESIDUAL VIBRATION REDUCTION ON A THREE MODE FLEXIBLE SYSTEM USING INPUT SHAPING

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Caracas, Venezuela, on August 10, 1976, the son of R. Ivan Blanco and Isolete De Almeida.

Education: Graduated from Stillwater High School, Stillwater, Oklahoma; received a B.S. degree from Oklahoma State University , Stillwater, Oklahoma, in 1999, in Mechanical Engineering. Completed the requirements for the Master of Science degree with a major in Mechanical Engineering with specialization in Dynamic Systems and Control Engineering at Oklahoma State University in August, 2001.

Experience: Research and Development Engineer at Mercruiser from June 2000 to present; Plant Layout Drafter at Mercruiser from March '99 to March '00; Engineering Liaison Intern at Boeing North American, Inc. Tulsa, OK from June '98 - August '98.

Professional Memberships: American Society of Mechanical Engineers and Certified as an Engineering Intern by the Oklahoma State Board of Registration for Professional Engineers and Land Surveyors.