

CR-189439

Minimizing Structural Vibrations with Input Shaping™

Final Report for Contract
NAS5-32034

Convolve, Inc.
January 1995

(NASA-CR-189439) MINIMIZING
STRUCTURAL VIBRATIONS WITH INPUT
SHAPING (TM) Final Report
(Convolve) 170 p

N95-30786

Unclass

G3/39 0058479

Contents

Background of the Technology.....	1
Summary of the Technology.....	2
Brief Description of the Drawings.....	5
Detailed Description of Developed Technologies.....	9
Extra-Insensitive and Specified Insensitivity Input Shapers.....	9
Introduction.....	9
Vector Diagrams.....	11
The Effects of Damping.....	12
Sensitivity to Errors in Natural Frequency.....	13
Effects of Modeling Errors on the Vector Diagram.....	13
Defining Insensitivity.....	14
Increasing Insensitivity by Relaxing Constraints.....	14
Experimental Results.....	16
Four-Impulse Extra-Insensitive Shapers.....	17
Extra-Insensitive Shapers with an Arbitrary Number of Humps in the Sensitivity Curve.....	18
Specified Insensitivity Shapers.....	20
Negative Input Shapers.....	22
Introduction.....	22
Over-Currenting with Negative Shapers.....	23
Negative ZV (Zero Vibration) Shapers.....	24
Negative ZVD (Zero Vibration & Derivative) Shapers.....	26
Negative EI (Extra-Insensitive) Shapers.....	26
Controlling High-Mode Excitation.....	28
Negative Shaper Experimental Results.....	30
Extra-Insensitive Input Shapers for Controlling Flexible Spacecraft.....	31
Introduction.....	31
Constraint Equation Derivation.....	33
Input Shaper Comparison.....	35
Input Shaper Verification.....	37
Multiple-Mode Impulse Shaping Sequences.....	39
Introduction.....	39
Two-Mode Sequences.....	39
Insensitivity to modeling errors.....	42
Experimental setup.....	42
Two-Mode Shaper Experimental Results.....	44
Input Shapers™ for Digital Systems.....	47
Introduction.....	47
Input Shaping Method.....	48
Digital Shaper Experimental results.....	49
Input Shaping™ for Trajectory Following.....	51
Introduction.....	51
Circular Trajectories.....	52
Square Trajectories.....	54
Improving Shaped Circular Responses.....	56
Improving Shaped Square Responses.....	56
Experimental Results.....	57
Using Different Shapers for Each Axis of Motion.....	58
Reduction of Vibration During Scanning.....	58
Whole Trajectory Shaping.....	58
Implementing Look-up Table for Shaped Trapezoidal Profiles on an LM628/629.....	59
Shaping Look-up Table for Arbitrary Commands.....	63
Shaping During Emergency Situations.....	64
Shaping Stepping Motor Systems.....	64
Shaping controller product.....	64

Tuning Trapezoidal, S-Curve, and Parabolic Profiles for Vibration Reduction	65
Introduction.....	65
Trapezoidal Velocity Profiles.....	65
S-Curve Velocity Profiles.....	67
Parabolic Profiled Trajectories.....	69
Procedures for Selecting the Parameters v_{max} , a_{max} , jerk for Trapezoidal Velocity Profiles.....	70
Procedures for Selecting the Parameters v_{max} , a_{max} , and jerk for S-Curve Velocity Profiles.....	71
Procedures for Selecting the Parameters v_{max} , a_{max} , and jerk for Parabolic Velocity Profiles.....	72
Input Shapers™ in Closed Loop Systems	72
Miscellaneous Shaping Applications.....	74
Figures.....	77
Appendix A - Manual for Input Shaping GAMS Scripts.....	116
Appendix B - MATLAB Input Shaping™ Toolbox	155

Executive Summary/ Commercialization Results

The result of this contract is a new method of commanding computer controlled machines to move with improved dynamic performance. This new technology enables machines to move quickly without spending time waiting for vibrations to settle. This contract generated significant research results. These results are now being commercialized for use within NASA and in the industrial sector.

For example, on March 2, 1995 the Space Shuttle Endeavor will be flying a payload called MACE (Mid Deck Active Control Experiment). A significant portion of the MACE experimental program consists of tests of control strategies that involve Input Shaping™. This experiment will validate the use of Input Shaping™ on space-based systems and structures.

On the commercial front, Convolve, Inc. is currently licensing Input Shaping™ into a variety of commercial products. As one example, Convolve, Inc. has signed a license to install Input Shaping™ on a machine that tests computer disk drive read/write heads for Seagate Technology's manufacturing operations. Over one hundred of these test machines are in service to perform 100% inspection of each head before they are assembled into computer disk drives. Convolve, Inc.'s software reduced the settle time of this testing machine from 200 msec to 23 msec. This improvement has significant ramifications for Seagate's production capacity since many moves are performed in order to test each head.

In addition to custom applications such as the Seagate machine, Convolve, Inc. is producing software products based on Input Shaping™ that run on standard controllers manufactured by two of the leading motion controller manufacturers, Motion Engineering and Delta Tau Data Systems. These software products consist of two parts. The first is a software tool that enables a user to monitor his system and configure the shaping algorithm. The second part is firmware for the motion controllers that implement the Input Shaping™ technique.

A third software product consists of an automated Input Shaper™ design tool that can be accessed through a computer dial-up service or through Email on the Internet. This service will use Convolve's highly optimized Input Shaping™ tools to custom-design multiple-mode Shaping™ sequences. The fourth tool that was commercialized from the results obtained under this NASA contract is a MATLAB Toolbox for Input Shaping™ design and analysis. The fifth tool is a series of GAMS optimization scripts for generating the most highly optimized Input Shapers™. These software tools are invaluable for research and new system development. The GAMS code and the MATLAB Toolbox have been supplied to NASA as a deliverable under this contract and are documented in this report.

Background of the Technology

This technology is a method for shaping the inputs to a dynamic system in order to minimize unwanted dynamics in that system.

Many physical systems must operate dynamically in order to accomplish their intended functions. However, in the course of their motions, the systems may acquire unwanted dynamics and vibrations which may be detrimental to their operation. For example, excessive vibrations in a dynamic system may result in larger than normal stresses and a premature failure of that system. Alternatively, if the system is designed to operate with a smooth, non-oscillatory motion, then vibrations may cause unwanted oscillations which actually prevent the system from achieving its intended purpose, or, at the very least, cause the system to operate at a significantly slower speed and lower performance level than originally intended. In addition, the unwanted dynamics may also degrade the performance of the system, either directly or indirectly, since the system is not exactly following its intended motion. As a result of these consequences, it is often desired to minimize the unwanted dynamics and vibrations in a physical system.

There are a number of approaches for achieving this end. One approach relies on altering the physical system in order to reduce any unwanted dynamics. For example, a robotic arm may be stiffened in order to reduce the amplitude of any residual vibrations, or the arm may be dampened so that residual vibrations quickly die out, or its mass be changed so that the resonant frequency of the arm is moved to a more favorable frequency. However, it is not always possible to alter the physical system. For example, the system may be so precisely designed as to be intolerant of the desired changes or it may just be physically impossible to alter the system, as is the case with a system which is inaccessible. Even if the system may be altered as desired, the alteration may come with a price - a more massive system, a larger actuator required to move the system or a more complex system, to name a few. Another approach relies on using a controller to actively reduce any unwanted dynamics. However, this approach also has its drawbacks. For example, most controllers rely on some sort of feedback from the unwanted dynamic and also on a good model of the system to be controlled, either of which may not always be available. In addition, controllers may be unacceptably complex, either in terms of the additional physical elements required to implement the controller or in terms of the time required for the controller to implement its control algorithm. In particular, fast real-time systems may be too fast for controllers to be an option.

A third approach, which is the approach considered by this technology, relies on altering the input to the system in order to reduce the unwanted dynamics. This approach

does not rely on physical alterations to the system, good models of the system to be controlled, or complex real-time calculations. In related work, Singer, et. al. [Singer, Neil C.; Seering, Warren P. "Preshaping Command Inputs to Reduce System Vibration". ASME Journal of Dynamic Systems, Measurement, and Control. (March 1990) and Singer, et. al. U.S. patent # 4,916,635, April 10, 1990] showed that residual vibration can be significantly reduced by employing an Input Shaping™ method that uses a simple system model and requires very little computation. The model consists only of estimates of the system's natural frequency and damping ratio. Constraints on the system inputs result in zero residual vibration if the system model is exact. When modeling errors exist, the shaped inputs keep the residual vibration of the system at a low level that is acceptable for many applications. Extending the method to systems with more than one modeled resonant frequency is straightforward[Singer, Neil C. Residual Vibration Reduction in Computer Controlled Machines. Ph.D. Thesis, Massachusetts Institute of Technology. (Feb. 1989)].

The shaping method works in real time by convolving a desired input with a sequence of impulses to produce the shaped input function that reduces residual vibration. The impulse sequence used in the convolution is called an input shaper. Selection of the number and type of impulses, the impulse amplitudes and their locations in time determine the amplitude and characteristics of the residual vibration. U.S. patent # 4,916,635 discloses the basic concept of using input shapers, while the current report discloses a variety of input shapers designed to achieve specific goals. Since this report considers many different input shapers used for various purposes, the description of the technologies subdivided into sections, with each section devoted to a specific class of input shapers.

Summary of the Technology

The technology for generating an input to a dynamic system to minimize unwanted dynamics in the system response includes establishing expressions quantifying the unwanted dynamics. First constraints which bound the available input to the dynamic system and second constraints which bound the unwanted dynamics are established, and a solution which is used to generate the input and allows maximum variation in the physical system characteristics while still satisfying the first and second constraints is found. The physical system is controlled based on the input to the physical system, whereby unwanted dynamics are minimized.

In a second aspect of this technology, the method for generating an input to a dynamic system to minimize unwanted dynamics in the system response includes establishing expressions quantifying the unwanted dynamics. First constraints which bound the available input to the dynamic system and second constraints which bound the unwanted

dynamics are established, and a solution which is used to generate the input and minimizes the length of the solution while still satisfying the first and second constraints is found. The physical system is controlled based on the input to the physical system, whereby unwanted dynamics are minimized. In a related aspect of this technology, the first and second constraints are supplemented by a third constraint bounding variations in physical system characteristics, and a solution which is used to generate the input and minimizes the length of the solution while still satisfying the first, second and third constraints is found.

In a third aspect of this technology, the method for generating an input to a dynamic system to minimize unwanted dynamics in the system response includes establishing expressions quantifying the unwanted dynamics. First constraints which bound the available input to the dynamic system and second constraints on variation in system response with variations in the physical system characteristics are established, and a solution which is used to generate the input and minimizes the length of the solution while still satisfying the first and second constraints is found. The physical system is controlled based on the input to the physical system, whereby unwanted dynamics are minimized.

Assuming the physical system is characterized by one or more vibrational modes each of which has a natural frequency and damping coefficient, and the solution is a sequence of impulses, then sequence may be defined implicitly, explicitly, or approximately by a set of equations which is chosen according to the constraints being considered and the desired characteristics of the sought solution. The sequence of impulses is convolved with an unshaped input and the result of the convolution is used as input to the dynamic system, whereby the unwanted dynamics are minimized.

In another aspect of this technology, the method for generating an input to a dynamic system to minimize unwanted dynamics in a physical system response includes establishing constraints on a sequence of impulses which minimize the unwanted dynamics. A first sequence of impulses which satisfy these constraints is determined, and a second sequence of impulses, which are discretized in location, are then determined. The number of impulses and locations of the impulses for the second sequence is determined based on the first sequence, and the amplitudes of the impulses of the second sequence are determined to satisfy the constraints. The second sequence of impulses is used to generate the input and the physical system is controlled based on the input to the physical system, whereby unwanted dynamics are minimized.

This technology is also a method for generating an input to a dynamic physical system to reduce the deviation between the shape of a trajectory traversed by a point in the physical system and a pre-selected shape includes establishing constraints on the available inputs to the dynamic system to define a group of possible inputs and determining an impulse

sequence which eliminates unwanted dynamics in the physical system. The impulse sequence is convolved with each input in the group of possible inputs to determine a group of shaped inputs and the shaped input which minimizes the deviation between the shape of the actual trajectory and the pre-selected shape is determined. The physical system is controlled based on the shaped input which minimizes the deviation between the shape of the actual trajectory and the pre-selected shape, whereby the deviation is minimized.

In another aspect of this technology, a method for generating an input to a dynamic system to minimize unwanted dynamics in the physical system response comprises establishing expressions quantifying the unwanted dynamics. First constraints on the available inputs to the dynamic system are established in order to define a group of possible inputs, each input in the group of possible inputs is expressed as the combination of one or more primitive input trains, and the primitive input train which minimizes the unwanted dynamics is determined. The physical system is controlled using the input in the group of possible inputs which corresponds to the primitive input train which minimizes the unwanted dynamics, whereby unwanted dynamics are minimized. The primitive input trains are sequences of impulses, and the group of possible inputs are commonly used inputs, such as those which generate trapezoidal, s-curve, or parabolic velocity profiles.

In a final aspect of this technology, the method for shaping an arbitrary command input to a dynamic physical system to reduce unwanted dynamics in the physical system includes determining a first parameterization for the arbitrary command input and determining an impulse sequence which eliminates unwanted dynamics in the physical system. The convolution of the impulse sequence with the arbitrary command input is then expressed using a second parameterization, which is based on the first parameterization, and the input to the physical system is controlled based on the second parameterization, whereby unwanted dynamics in the physical system are minimized.

Brief Description of the Drawings

FIGS. 1a, 1b and 1c illustrate the convolution of an unshaped system input with a sequence of impulses to produce a shaped system input;

FIGS. 2a and 2b illustrate the correspondence between an impulse sequence and its vector diagram;

FIGS. 3a, 3b and 3c illustrate the correspondence between a vector diagram and its time domain representation of vibration;

FIGS. 4a and 4b illustrate the scaling effect of a damping coefficient;

FIG. 5a illustrates a vector diagram for a three-impulse sequence;

FIG. 5b graphs residual vibration versus normalized modeling error for the three-impulse sequence of FIG. 5a;

FIG. 6a illustrates a vector diagram for an asymmetric three-impulse sequence;

FIG. 6b graphs residual vibration versus normalized modeling error for the three-impulse sequence of FIG. 6a;

FIG. 7 graphs residual vibration versus normalized modeling error for two different three-impulse sequences, and illustrates the allowable normalized modeling error for two different values of insensitivity;

FIG. 8 is a graph of table position versus time for shaped and unshaped inputs;

FIG. 9 is a graph of the experimental and theoretical residual vibration versus normalized modeling error for the case of $V=0.05$;

FIG. 10 is a graph of the experimental and theoretical residual vibration versus normalized modeling error for the case of $V=0.10$;

FIG. 11 graphs residual vibrations versus normalized frequency for a four-impulse sequence;

FIG. 12 graphs the amplitude of the first impulse of a multi-impulse sequence as a function of insensitivity and damping coefficient;

FIG. 13 graphs the location of the second impulse of a multi-impulse sequence as a function of insensitivity and damping coefficient;

FIGS. 14a, 14b and 14c illustrate over-currenting caused by impulse sequences containing negative impulses;

FIGS. 15, 16 and 17 graph the residual vibration versus normalized frequency for different impulse sequences;

FIG. 18 graphs the length of an impulse sequence versus the peak partial sum;

FIG. 19 graphs the residual vibration versus normalized frequency for three different impulse sequences with a peak partial sum of one;

FIGS. 20 and 21 graph the residual vibration versus normalized frequency for two

different impulse sequences;

FIG. 22 graphs the residual vibration versus normalized frequency for an impulse sequence used with a low-pass filter;

FIG. 23 graphs encoder position versus time for shaped and unshaped inputs;

FIG. 24 graphs encoder position versus time for four different impulse sequences;

FIG. 25 illustrates a simple system model;

FIGS. 26a, 26b and 26c illustrate the convolution of a system input with a sequence of impulses to produce a shaped system input of constant magnitude;

FIG. 27 graphs residual vibration versus frequency for different impulse sequences;

FIGS. 28a and 28b graph position versus time for various impulse sequences and spring constants;

FIG. 29 graphs magnitude versus frequency for the vibration resulting from an unshaped input;

FIG. 30 graphs table position versus time for an unshaped and a shaped input;

FIG. 31 illustrates a symmetric, five impulse sequence which reduces vibrations for two different modes;

FIG. 32 graphs the residual vibration versus system frequency for a sequence of the type shown in FIG. 31;

FIG. 33 illustrates the CMS system;

FIG. 34 graphs the residual vibration versus frequency for a nine-impulse and fifteen-impulse sequence;

FIG. 35 graphs encoder position versus time;

FIGS. 36a, 36b, 36c, and 36d graph laser derived position versus time for four different impulse sequences;

FIG. 37 graphs the magnitude of the FFT versus frequency for a shaped and an unshaped input;

FIG. 38 illustrates a situation in which a three-impulse sequence does not fit the discrete spacing of a system;

FIG. 39 graphs residual vibrations versus system frequency for four different impulse sequences;

FIG. 40a illustrates how a three-impulse sequence which does not fit the discrete spacing of a system;

FIG. 40b illustrates a five-impulse sequence, based on the three-impulse sequence of FIG. 40a, which does fit the discrete spacing of a system;

FIG. 41 illustrates an experimental configuration with two modes of vibration;

FIGS. 42, 43, 44 and 45 graph table position as a function of time for various inputs;

FIG. 46 illustrates a two-mode model for a flexible system under PD control;

FIG. 47 graphs the trajectories resulting from an unshaped and a shaped input;

FIG. 48 graphs the command amplitude versus time for four different commands;

FIG. 49 graphs radius envelope versus vibration for different inputs and damping coefficients;

FIG. 50 graphs radius envelope versus departure angle for various inputs;

FIG. 51 graphs radius envelope versus frequency for various inputs;

FIG. 52 graphs the trajectories resulting from an unshaped and a shaped input;

FIG. 53 graphs the command amplitude versus time for various inputs;

FIG. 54 graphs mean error versus cycles per square for various input and damping coefficients;

FIG. 55 graphs the trajectories resulting from an unshaped and a shaped input for a departure angle of thirty degrees;

FIG. 56 graphs mean error versus departure angle for various inputs;

FIG. 57 graphs mean error versus frequency for various inputs;

FIG. 58 graphs the trajectory resulting from using a shaped input designed for a circle of increased radius and allowing for transients;

FIG. 59 graphs tracking error versus time for a shaped input used to trace a square;

FIG. 60 illustrates a device used to record different trajectories;

FIG. 61 illustrates the trajectory produced by an unshaped input;

FIG. 62 illustrates the trajectory produced by a shaped input;

FIG. 63 illustrates the trajectory produced by an unshaped input when an endpoint mass is added to the device of FIG. 60;

FIG. 64 illustrates the trajectory produced by a shaped input when an endpoint mass is added to the device of FIG. 60;

FIG. 65 graphs velocity versus time for an unshaped and shaped trapezoidal velocity profile;

FIGS. 66a and 66b illustrate that the superposition of four shaped acceleration steps yields a shaped trapezoidal velocity profile;

FIG. 67 graphs pullout torque versus step rate for a stepper motor;

FIGS. 68a and 68b illustrate a trapezoidal velocity profile and its corresponding acceleration profile;

FIGS. 69a, 69b, 69c and 69d illustrate that the acceleration profile of FIG. 68b may be decomposed into the convolution of a step function with two sequences of two impulses each;

FIGS. 70 and 71 graph residual vibration versus frequency for different inputs;

FIGS. 72a, 72b and 72c illustrate an s-curve velocity profile and its corresponding acceleration and jerk profiles;

FIGS. 73a, 73b, 73c, 73d and 73e illustrate that the jerk profile of FIG. 72c may be decomposed into the convolution of a step function with three sequences of two impulses each;

FIG. 74 graphs residual vibration versus frequency for different inputs; and

FIGS. 75a, 75b and 75c illustrate a parabolic velocity profile and its corresponding acceleration and jerk profiles;

FIG. 76 is a block diagram of a closed loop system with an internal input shaper;

FIG. 77 is a block diagram of a closed loop system with an internal input shaper and a compensating filter;

FIG. 78 is a block diagram of a closed loop system with an external input shaper;

FIG. 79 is a schematic illustration of the technology disclosed herein.

Detailed Description of Developed Technologies

Extra-Insensitive and Specified Insensitivity Input Shapers

Introduction

An early pre-cursor of Input Shaping™ was the use of posicast control by O.J.M. Smith [Smith, O.J.M. Feedback Control Systems. McGraw-Hill Book Company, Inc., New York (1958)]. This technique breaks a step input into two smaller steps, one of which is delayed in time. The result is a reduced settling time for the system. Wiederrich and Roth [Wiederrich, J. L.; Roth B., "Dynamic Synthesis of Cams Using Finite Trigonometric Series". Journal of Engineering for Industry (Feb. 1975)] shaped cam profiles to control the harmonic content of the imposed vibration. Their methods reduced steady-state vibration and assured the accuracy of their simple model.

Optimal control approaches have been used to generate input profiles for commanding vibratory systems. Junkins, Turner, Chun, and Juang have made considerable progress toward practical solutions of the optimal control formulation for flexible systems [Junkins, John L.; Turner, James D. "Optimal Spacecraft Rotational Maneuvers". Elsevier Science Publishers, New York. (1986) , Chun, Hon M.; Turner, James D.; Juang, Jer-Nan. "Disturbance-Accommodating Tracking Maneuvers of Flexible Spacecraft". Journal of the Astronautical Sciences 33, 2. (April-June, 1985)]. Gupta [Gupta, Narendra K. "Frequency-Shaped Cost Functionals: Extension of Linear-Quadratic". Journal of Guidance and Control 3, 6 (Nov.-Dec, 1980)], and Junkins and Turner [Junkins, John L.; Turner, James D. "Optimal Spacecraft Rotational Maneuvers". Elsevier Science Publishers, New York. (1986)] included frequency shaping terms in their optimal formulation.

Farrenkopf [Farrenkopf, R. L. "Optimal Open-Loop Maneuver Profiles for Flexible Spacecraft". Journal of Guidance and Control 2, 6. (Nov.-Dec., 1979)] developed velocity shaping techniques for flexible spacecraft. Swigert [Swigert, C. J. "Shaped Torque Techniques". Journal of Guidance and Control 3, 5 (Sep.-Oct., 1980)] demonstrated that torque shaping can be implemented on systems which modally decompose into second-order harmonic oscillators.

Singer and Seering [Singer, Neil C.; Seering, Warren P. "Preshaping Command Inputs to Reduce System Vibration". ASME Journal of Dynamic Systems, Measurement, and Control. (March 1990)] showed that residual vibration can be significantly reduced by employing an Input Shaping™ method that uses a simple system model and requires very little computation. The model consists only of estimates of the system's natural frequency

and damping ratio. Constraints on the system inputs result in zero residual vibration if the system model is exact. When modeling errors exist, the shaped inputs keep the residual vibration of the system at a low level that is acceptable for many applications. There is a straightforward manner to extend this method to systems with more than one modeled resonant frequency [Singer, Neil C. Residual Vibration Reduction in Computer Controlled Machines. Ph.D. Thesis, Massachusetts Institute of Technology. (Feb. 1989)].

The shaping method works in real time by convolving a desired input with a sequence of impulses to produce the shaped input function that reduces residual vibration. The impulse sequence used in the convolution is called an input shaper. For example, if it is desired to move a system from one point to another, (a step change in position) then the convolution of the step function with a sequence of impulses results in a shaped input which is a series of steps, or a staircase. Similarly, if a constant ramp input is commanded, the shaped input will be a ramp whose slope changes value as a function of time. Instead of giving the system the step or ramp input, the system is given the shaped input. Selection of the impulse amplitudes and locations in time determine the amplitude of residual vibration.

The commanded input is not limited to steps and ramps. Rather, any command function can be shaped with an impulse sequence. FIG. 1a depicts a general input which is convolved with the impulse sequence of FIG. 1b. The resulting shaped input is shown as the solid line in FIG. 1c; while the dotted and dashed lines are the two components arising from each of the two impulses of FIG. 1b. Sequences containing three impulses have been shown to yield particularly effective system inputs (when convolved with system commands) both in terms of vibration suppression and response time. The shaping method is effective in reducing vibration in both open and closed loop systems.

This section extends the basic Input Shaping™ technique of Singer and Seering and concentrates on generating different impulse sequences to be used in the convolution that produces the vibration-reducing inputs. Unlike the time domain analysis presented in references [Singer, Neil C. Residual Vibration Reduction in Computer Controlled Machines. Ph.D. Thesis, Massachusetts Institute of Technology. (Feb. 1989), Singer, Neil C.; Seering, Warren P. "Preshaping Command Inputs to Reduce System Vibration". ASME Journal of Dynamic Systems, Measurement, and Control. (March 1990)], this work uses vector diagrams, which are graphical representations of impulse sequences, to generate and evaluate the vibration-reducing characteristics of impulse sequences. By relaxing the constraints used by Singer and Seering, a variety of sequences can be generated that give better performance than those reported previously.

Vector Diagrams

To understand the results presented in this section, one must be familiar with the vector diagram representation of vibration and its use in creating shaped inputs. An explanation of vector diagram methods was presented in [Singhose, William E. "Shaping Inputs to Reduce Residual Vibration: a Vector Diagram Approach". MIT Artificial Intelligence Lab Memo No. 1223. (March, 1990)] and will be summarized here.

A vector diagram is a graphical representation of an impulse sequence in polar coordinates (r - θ space). A vector diagram is created by setting r equal to the amplitude of an impulse and by setting $\theta = \omega T$, where ω (rad/sec) is a chosen frequency and T is the time location of the impulse. FIG. 2a shows a typical impulse sequence and FIG. 2b shows the corresponding vector diagram.

Vector diagrams become useful tools for producing vibration-reducing impulse sequences when ω is set equal to the best estimate of a natural frequency, ω_{sys} , of a system and the time of the first impulse is set to zero ($T_1 = 0$). When a vector diagram is created in this manner, the amplitude of the resultant, A_R , is proportional to the amplitude of residual vibration of a system driven by a step convolved with the impulse sequence. The angle of the resultant is the phase of the vibration relative to the system response to an impulse at time zero.

Because arbitrary inputs can be built as sums of steps, the amplitude of A_R is a measure of system response to arbitrary inputs. This result enables us to determine residual vibration geometrically; the residual vibration is calculated by geometrically summing the vectors on the vector diagram. FIG. 3a shows the vector diagram representation for the resultant vibration of a second order undamped system. FIG. 3b shows the time domain representation of each component of the vibration, and FIG. 3c shows the time domain representation of the resultant. On a vector diagram, vibration appears as a vector, whereas, in the time domain, vibration appears as a sinusoid.

We can use the vector diagram to generate impulse sequences that yield vibration-free system response. To do this, we place n arbitrary vectors on a vector diagram and then cancel the resultant of the first n vectors with an $n+1$ st vector. When the $n+1$ vectors are converted into an impulse sequence, and the sequence is convolved with a desired system input, the resulting shaped input will cause no residual vibration when applied to a system with natural frequency ω . Additionally, if the sum of impulse amplitudes is normalized to one, the system will stop at the commanded setpoint.

The magnitude, A_{n+1} , and angle, θ_{n+1} , of the canceling vector are given by:

$$|A_{n+1}| = \sqrt{|R_x|^2 + |R_y|^2} \quad \theta_{n+1} = \pi + \tan^{-1} \left\{ \frac{R_y}{R_x} \right\} \quad (1)$$

where R_x and R_y are the horizontal and vertical components of the resultant. These components are given by

$$R_x = \sum_{i=1}^n A_i \cos \theta_i \quad R_y = \sum_{i=1}^n A_i \sin \theta_i \quad (2)$$

where A_i and θ_i are the magnitude and angles of the n vectors to be cancelled.

The Effects of Damping

When the system has viscous damping, the vector diagram representation of vibration must be modified in two ways. First, we must use the damped natural frequency for plotting the vector diagram. This corresponds to using:

$$\theta = \sqrt{1 - \zeta^2} \omega T \quad (3)$$

where ζ is the damping coefficient.

Second, the amplitudes of the vectors must be scaled to account for damping. As time progresses, the amplitude of response decays; therefore, the amplitude of the canceling vector decreases. For example, if we give a system an impulse with amplitude A_1 , at time zero, the single impulse, A_2 , that will cancel the system's vibration is located π radians (180°) out of phase with the first impulse, but it has a smaller amplitude as shown in FIGS. 4a and 4b. If a system has a damping ratio of ζ , then the amplitude of the second impulse is:

$$A_2 = A_1 e^{-\zeta \omega T} = A_1 e^{-\zeta' \theta} \quad (4)$$

where θ is given by Eq. 3 and

$$\zeta' = \zeta / (1 - \zeta^2)^{1/2} \quad (5)$$

We define the effective amplitude, $|A_{eff}|$, of an impulse, A , occurring at time T to be the amplitude of an impulse occurring at time zero whose vibratory response would decay to the amplitude of response caused by A at time T . Written in equation form, the effective amplitude of a vector is:

$$|A_{eff}| = \frac{|A|}{e^{-\zeta' \theta}} \quad (6)$$

When we cancel n vectors with an $n+1$ st vector on a vector diagram to create a vibration-eliminating impulse sequence, we must use Eq. 3 to determine the angles of the vectors and assign each of the n vectors an effective amplitude according to Eq. 6 before using Eqs. 1 to solve for the $n+1$ st vector. When we include the effects of damping, the equations describing the $n+1$ st canceling vector are:

$$|A_{n+1}| = (e^{-\zeta' \theta_{n+1}}) \sqrt{|R_x|^2 + |R_y|^2} \quad \theta_{n+1} = \pi + \tan^{-1} \left\{ \frac{R_y}{R_x} \right\} \quad (7)$$

where R_x and R_y are given by:

$$R_x = \sum_{i=1}^n A_{ieff} \cos \theta_i \quad R_y = \sum_{i=1}^n A_{ieff} \sin \theta_i \quad (8)$$

Sensitivity to Errors in Natural Frequency

It is possible to create an infinite number of vibration-reducing input functions with the vector diagram tool. The "best" would seem to be the one that works most effectively on real systems. Because there will always be some error in the estimate of natural frequency for any system, the sensitivity of the shaped input to modeling errors is important. When the system model is not exact, some residual vibration will occur when the system is moved with the shaped inputs. A plot of the vibration versus error in estimated natural frequency for a three-impulse sequence developed by Singer and Seering is shown in FIG. 5b and the corresponding vector diagram is shown in FIG. 5a. This impulse sequence produces a system response that is fairly insensitive to errors or changes in the system parameters. That is, there is relatively little vibration in the system even when the resonant frequency estimate is off by 15% as shown.

Effects of Modeling Errors on the Vector Diagram

The sensitivity curve shown in FIG. 5b can be obtained directly from a vector diagram if we analyze how a modeling error changes the diagram. When the natural frequency of a system differs from the assumed natural frequency, the error can be represented on a vector diagram by shifting each vector through an angle ϕ . If ω_{sys} is the actual natural frequency of the system and ω is the modeling frequency, then the error in frequency is $\omega - \omega_{sys}$. The angle through which the vectors are shifted, ϕ , is related to the frequency error by the equation:

$$\phi = (\omega - \omega_{sys})T \quad (9)$$

The error in modeling causes a non-zero resultant to be formed on the vector diagram if the vectors were determined by Eqs. 1. The resultant that is formed represents the vibration that is induced by the error in frequency.

Given that modeling errors cause a resultant, R_{err} , on a vector diagram, we can compare the sensitivity of different input functions to modeling errors by plotting the amplitude of R_{err} versus the error in frequency. If we plot a sensitivity curve like the one shown in FIG. 5b, we can determine how much vibration will result from a given error in estimated frequency. To make a sensitivity curve, we must develop an expression for the amplitude of the resultant as a function of the error in frequency ($\omega - \omega_{sys}$). This has been done previously [Singhose, William E. "A Vector Diagram Approach to Shaping Inputs for Vibration Reduction". MIT Artificial Intelligence Lab Memo No. 1223. (March, 1990)],

and the relation is:

$$|R_{\text{err}}| = \sqrt{|R_{x\text{err}}|^2 + |R_{y\text{err}}|^2} \quad (10)$$

where:

$$R_{x\text{err}} = \sum_{i=1}^n A_{i\text{eff}} \cos(\theta_i - \phi_i) \quad R_{y\text{err}} = \sum_{i=1}^n A_{i\text{eff}} \sin(\theta_i - \phi_i) \quad (11)$$

$$A_{i\text{eff}} = \frac{A_i}{e^{-\zeta(\theta_i - \phi_i)}} \quad (12)$$

$$\phi_i = \frac{\omega - \omega_{\text{sys}}}{\omega} \theta_i \quad (13)$$

Defining Insensitivity

We define the insensitivity of a sequence to be the width of the sensitivity curve at a given level of residual vibration. If the acceptable level of vibration is 5% of the vibration resulting from an unshaped input, then we draw a horizontal line across the sensitivity curve at 5% as shown by the dashed line in FIG. 5b, and the distance between the points of intersection is the insensitivity. Quantitatively, the insensitivity is defined as $I = \Delta\omega/\omega_{\text{sys}}$, where $\Delta\omega$ is the distance between the two points of intersection. For example, the insensitivity of the impulse sequence represented by the vector diagram of FIG. 5a is 0.286, because the impulse sequence associated with the sensitivity curve of FIG. 5b causes less than 5% of the unshaped vibration from $(\omega/\omega_{\text{sys}})_{\text{lo}} = 0.857$ to $(\omega/\omega_{\text{sys}})_{\text{hi}} = 1.143$.

Increasing Insensitivity by Relaxing Constraints

The sensitivity curve in FIG. 5b can be widened by displacing the vectors from the horizontal axis, that is by not placing the second vector at π or the third vector at 2π . When the vectors are located off the horizontal axis, the sensitivity curve is skewed; it is not symmetrical about $\omega/\omega_{\text{sys}} = 1$. For example, we can modify the vector diagram in FIG. 5a by placing the second vector at an angle of 154 degrees, keeping the amplitude fixed at 2, and still satisfying Eqs. 1, as shown in FIG. 6a. The sensitivity curve for this sequence is shown in FIG. 6b. The insensitivity for this input function is 0.408 (0.93 to 1.338), a 43% improvement over that of FIG. 5b. An interesting feature to note is that the sensitivity curve is skewed to the right, i.e., it is more insensitive to errors that are higher in frequency than the modeling frequency. This may be a desirable property of an input function if the system being moved increases its natural frequency during some part of its operation. However, an approximation of a system's resonant frequency is usually as likely to be too high as too low, so it seems desirable to maintain symmetrical insensitivity in most cases.

To increase insensitivity and still maintain a symmetrical sensitivity curve, we can relax Singer's constraint of zero vibration when the system model is exact. Any system model will have some amount of approximation, so giving up this strict constraint is reasonable. Insensitivity is increased significantly if we reformulate the constraints in the following way. Pick an allowable level of residual vibration, V . Then, calculate amplitudes for the impulses at $\theta = 0, \pi$, and 2π such that the residual vibration equals V when $\omega/\omega_{\text{sys}} = 1$. Furthermore, require the sensitivity curve to drop to zero on either side of $\omega/\omega_{\text{sys}} = 1$.

From the above conditions, we can derive the three-impulse sequence that yields the maximum insensitivity for a given vibration limit. The sensitivity curve will be constrained to be symmetrical about the modeling frequency, this means the angle of the third vector, θ_3 , is always twice the angle of the second vector, θ_2 . In equation form:

$$\theta_3 = 2\theta_2 \quad (14)$$

When the resultant at the modeling frequency is set equal to the vibration limit, V , we have:

$$|A_1| - |A_2| + |A_3| = V(|A_1| + |A_2| + |A_3|) \quad (15)$$

The value of $|A_2|$ is subtracted from the left side of Eq. 15 because the vector A_2 points in the opposite direction of A_1 and A_3 on the vector diagram. We have arbitrarily set $|A_1|$ equal to one, so Eq. 15 reduces to:

$$|A_2| = \frac{(1 - V)(1 + |A_3|)}{(1 + V)} \quad (16)$$

Because we are forcing the sensitivity curve to drop to zero on either side of $\omega/\omega_{\text{sys}} = 1$, the resultant on the vector diagram must equal zero for the values of $\theta + \alpha$ and $\theta - \beta$, where α and β are some unknown deviations from the angle corresponding to $\omega/\omega_{\text{sys}} = 1$.

In equation form this constraint is:

$$0 = 1 + |A_2|\cos(\theta_2 + \alpha) + |A_3|\cos(\theta_3 + 2\alpha) \quad (17)$$

$$0 = |A_2|\sin(\theta_2 + \alpha) + |A_3|\sin(\theta_3 + 2\alpha) \quad (18)$$

$$0 = 1 + |A_2|\cos(\theta_2 - \beta) + |A_3|\cos(\theta_3 - 2\beta) \quad (19)$$

$$0 = |A_2|\sin(\theta_2 - \beta) + |A_3|\sin(\theta_3 - 2\beta) \quad (20)$$

Eqs. 14 and 16-20 are six equations with seven unknowns, ($A_2, A_3, \theta_2, \theta_3, \alpha, \beta$ and V). Neglecting damping and solving the equations in terms of V , we obtain the following values for the extra-insensitive sequence we were seeking:

$$\begin{aligned} |A_1| &= \frac{1+V}{4} & |A_2| &= \frac{1-V}{2} & |A_3| &= \frac{1+V}{4} \\ \theta_1 &= 0 & \theta_2 &= \pi & \theta_3 &= 2\pi \end{aligned} \quad (21)$$

If we examine the sensitivity curves for the above sequence, we discover that when the vibration limit is increased, insensitivity improves significantly. The insensitivity is

0.399 when $V = 0.05$, a 39% improvement over Singer's three-impulse sequence of FIG. 5a. The insensitivity further increases from 0.399 to 0.561 when V is increased from 0.05 to 0.10, as shown in FIG. 7.

When the system has viscous damping, the constraint equations cannot be solved in closed form. Fortunately, the equations containing damping terms that are analogous to Eqs. 14-20 can be solved numerically to obtain the impulse sequence as a function of two variables, ζ and V . Numerical solutions were calculated for $0 \leq \zeta \leq 0.3$ and $0 \leq V \leq 0.15$. A surface was fit to the data and the following description of the extra-insensitive sequence in the time domain was obtained:

$$\begin{aligned}
 A1 &= 0.2497 + 0.2496V + 0.8001\zeta + 1.233V\zeta + 0.4960\zeta^2 + 3.173V\zeta^2 \\
 A2 &= 1 - (A1 + A3) \\
 A3 &= 0.2515 + 0.2147V - 0.8325\zeta + 1.415V\zeta + 0.8518\zeta^2 - 4.901V\zeta^2 \\
 T1 &= 0 \\
 T2 &= (0.5000 + 0.4616V\zeta + 4.262V\zeta^2 + 1.756V\zeta^3 \\
 &+ 8.578V^2\zeta - 108.6V^2\zeta^2 + 337.0V^2\zeta^3) T_d \\
 T3 &= T_d
 \end{aligned} \tag{22}$$

where, T_d is the period of damped vibration:

$$T_d = \frac{2\pi}{\omega \sqrt{1-\zeta^2}} \tag{23}$$

If we examine the sensitivity curves for the above sequence, we discover that damping increases the insensitivity to modeling errors. For example, the undamped sequence based on $V = 5\%$ has an insensitivity of 0.399 while the sequence corresponding to $V = 5\%$ and $\zeta = 0.1$ has an insensitivity of 0.470, an 18% improvement. When the damped sequence is used, the frequencies at which the sensitivity goes to zero are more distant from the modeling frequency and the sensitivity curves are skewed toward the higher frequencies.

Experimental Results

Tests were performed on the assembly robot described in reference [Vaaler, Erik; Seering, Warren P., "Design of a Cartesian Robot". Presented at the winter annual meeting of the ASME (1986)]. A steel beam with a mass at one end was attached to a turntable on the robot base. The table was driven by a DC motor under PD control and its position was measured by an optical encoder.

When the table was given a step input in position, large oscillations were induced in the beam-mass system. The solid line of FIG. 8 shows a typical system response to a step input. The system parameters were determined by examining the data from a step

response. The natural frequency was estimated to be 2.8 Hertz and the damping ratio was approximated as 0. The step input was then shaped by the impulse sequence given by Eqs. 21 with the vibration limit set to 5%. The dotted curve in FIG. 8 shows the system response to the shaped input.

Sensitivity curves for the impulse sequence were experimentally determined by purposely introducing errors in the system model. The experimentally determined natural frequency was chosen as the "exact" frequency ($\omega_{\text{SYS}} = 2.8$ Hz). Impulse sequences were then derived for frequencies ranging from $0.7\omega_{\text{SYS}}$ to $1.4\omega_{\text{SYS}}$ (1.96 Hz - 3.92 Hz). Each impulse sequence was used to shape a step command and the shaped input was applied to the system. The amplitude of the resulting vibration was recorded. This amplitude was then divided by the baseline value to get the percentage of the unshaped vibration caused by the shaped input. By plotting the percentage of the unshaped vibration versus the normalized modeling error ($\omega/\omega_{\text{SYS}}$), sensitivity curves were obtained.

FIG. 9 shows the experimentally determined sensitivity data when the vibration limit was set to 5%. The data follows the same general shape as the theoretically determined sensitivity curve of FIG. 7. The curve has a non-zero value when the system model is correct and it slopes down toward zero on either side of the modeling frequency. The experimentally determined insensitivity for $V = 0.05$ was approximately 0.42, which slightly exceeds the theoretical insensitivity of 0.399. FIG. 10 shows the experimental sensitivity curve for $V = 0.1$. The insensitivity increased to approximately 0.67, once again better than the theoretical insensitivity for $V = 0.1$ of 0.561. The experimental results were somewhat better than the theory predicted because the system model was undamped, while the actual hardware did have some small amount of damping. As was previously remarked, damping increases insensitivity to modeling errors.

Four-Impulse Extra-Insensitive Shapers

Instead of limiting the sequence to three impulses, we can obtain more insensitivity with a four-impulse sequence that has a length of 1.5 periods of vibration. To maximize the insensitivity for a shaper of this length, we solve the following equations for a damped system:

$$\sum A_i = 1 \tag{24}$$

$$0 = e^{-\zeta\omega t} \{ (\sum A_i e^{\zeta\omega t} \cos(\omega\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\omega t} \sin(\omega\sqrt{1-\zeta^2}t_i))^2 \}^{1/2} \tag{25}$$

$$V = e^{-\zeta\omega t} \{ (\sum A_i e^{\zeta\omega t} \cos(\omega\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\omega t} \sin(\omega\sqrt{1-\zeta^2}t_i))^2 \}^{1/2} \tag{26}$$

$$\begin{aligned}
0 &= \frac{d}{d\omega_{l0}} \left(e^{-\zeta\omega_{l0}t_4} \left\{ (\sum A_i e^{\zeta\omega_{l0}t_i} \cos(\omega_{l0}\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\omega_{l0}t_i} \sin(\omega_{l0}\sqrt{1-\zeta^2}t_i))^2 \right\}^{1/2} \right) \\
0 &= \frac{d}{d\omega_{hi}} \left(e^{-\zeta\omega_{hi}t_4} \left\{ (\sum A_i e^{\zeta\omega_{hi}t_i} \cos(\omega_{hi}\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\omega_{hi}t_i} \sin(\omega_{hi}\sqrt{1-\zeta^2}t_i))^2 \right\}^{1/2} \right)
\end{aligned} \tag{27}$$

$$\begin{aligned}
0 &= e^{-\zeta\omega_{l2}t_4} \left\{ (\sum A_i e^{\zeta\omega_{l2}t_i} \cos(\omega_{l2}\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\omega_{l2}t_i} \sin(\omega_{l2}\sqrt{1-\zeta^2}t_i))^2 \right\}^{1/2} \\
0 &= e^{-\zeta\omega_{h2}t_4} \left\{ (\sum A_i e^{\zeta\omega_{h2}t_i} \cos(\omega_{h2}\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\omega_{h2}t_i} \sin(\omega_{h2}\sqrt{1-\zeta^2}t_i))^2 \right\}^{1/2}
\end{aligned} \tag{28}$$

where, ω_{hi} and ω_{h2} are unknown (variables) frequencies that are higher than ω . ω_{l0} and ω_{l2} are unknown frequencies lower than ω , ω_{h2} is higher than ω_{hi} , and ω_{l2} is lower than ω_{l0} . Note that each of the eqs. in 27-28 produce two more equations in the optimization process, one for the sin part and one for the cos part.

The solution to Eqs. 24-28 when $\zeta=0$ is:

$$\begin{aligned}
A_1 &= \frac{3x^2 + 2x + 3V^2}{16x} & A_2 &= 0.5 - \frac{3x^2 + 2x + 3V^2}{16x} & A_3 &= 0.5 - \frac{3x^2 + 2x + 3V^2}{16x} & A_4 &= \frac{3x^2 + 2x + 3V^2}{16x} \\
T_1 &= 0 & T_2 &= 0.5T & T_3 &= T & T_4 &= 1.5T
\end{aligned} \tag{29}$$

where T is the period of vibration and:

$$x = \{ \sqrt{2} [(1-V^2)^{1/2} + 1] \}^{1/3}$$

A sensitivity curve for the four-impulse extra-insensitive sequence is shown in FIG. 11. When damping is added to the problem formulation, the equations for the four-impulse extra-insensitive shaper cannot be solved in closed form. However, the equations can be solved numerically to obtain the impulse sequence as a function of two variables, ζ and V. Numerical solutions were calculated for $0 \leq \zeta \leq 0.2$ and $V=0.05$. A curve was fit to the data and the following description of the four-impulse extra-insensitive shaper for $V=0.05$ was obtained as:

$$\begin{aligned}
A_1 &= 0.1608 + 0.7475 \zeta + 1.948 \zeta^2 - 0.4882 \zeta^3 \\
A_2 &= 1 - (A_1 + A_3 + A_4) \\
A_3 &= 0.3394 - 0.5466 \zeta - 1.1354 \zeta^2 + 2.6167 \zeta^3 \\
A_4 &= 0.1589 - 0.5255 \zeta + 0.4152 \zeta^2 + 1.0164 \zeta^3 \\
T_1 &= 0 \\
T_2 &= (0.5000 + 0.1426 \zeta - 0.6243 \zeta^2 + 6.590 \zeta^3) T_d \\
T_3 &= (1.0 + 0.17226 \zeta - 1.725 \zeta^2 + 10.058 \zeta^3) T_d \\
T_4 &= T_d
\end{aligned} \tag{30}$$

Extra-Insensitive Shapers with an Arbitrary Number of Humps in the Sensitivity Curve

The EI formulation can be extended to obtain any number of humps in the sensitivity curve.

The three-impulse EI shaper had one hump and the four-impulse shaper had two humps. The constraint formulation for the Q-hump EI shaper can be summarized as follows:

For Q even,

- 1) Set the vibration equal to 0 at the modeling frequency.
- 2) Set the vibration equal to V and the derivative equal to 0 at $Q/2$ frequencies higher than the modeling frequencies and $Q/2$ frequencies lower than the modeling frequency.
- 3) Set the vibration equal to 0 at $Q/2$ frequencies higher than the modeling frequencies and $Q/2$ frequencies lower than the modeling frequency.
- 4) The frequencies in steps 2 and 3 must alternate as we go away from the modeling frequency, with a frequency from step 2 occurring first.

For Q odd,

- 1) Set the vibration equal to V and the derivative equal to 0 at the modeling frequency.
- 2) Set the vibration equal to 0 at $(Q+1)/2$ frequencies higher than the modeling frequencies and $(Q+1)/2$ frequencies lower than the modeling frequency.
- 3) Set the vibration equal to V and the derivative equal to 0 at $(Q-1)/2$ frequencies higher than the modeling frequencies and $(Q-1)/2$ frequencies lower than the modeling frequency.
- 4) The frequencies in steps 2 and 3 must alternate as we go away from the modeling frequency, with a frequency from step 2 occurring first.

Specified Insensitivity Shapers

The previously described shapers were derived by specifying the shaper time length and then maximizing the insensitivity. As an alternative approach to shaper design, the desired insensitivity to modeling errors can be specified and then the shaper can be solved for by minimizing the length. The difficulty with this process is that the set of equations to be solved depends on the desired insensitivity. This can be seen by examining the undamped three and four-impulse extra-insensitive shapers presented above. (These shapers can be considered specified insensitivity shapers whose desired insensitivities are 0.399 and 0.726, respectively.) The equations for the three-impulse EI, Eqs. 14, 16-20, are noticeably different than the equations for the four-impulse EI shaper, Eqs. 24-28.

The equations describing an SI shaper can be determined from the desired insensitivity and the system damping. The first step in formulating the set of constraint equations is to determine how many humps there will be in the sensitivity curve. For instance, the three-impulse EI shaper has 1 hump in its sensitivity curve, while the four-impulse EI shaper has two humps, as shown in FIGS. 7 and 11. Given the desired insensitivity, I , the number of humps can be determined for $V=0.05$ from the following decision tree:

$$\begin{aligned} &\text{if } I < 0.2218 + 0.3143 \zeta + 0.1819 \zeta^2 + 0.4934 \zeta^3, \\ &\quad \text{then the number of humps} = 0; \\ &\text{if } I < 0.5916 + 0.7647\zeta + 0.60\zeta^2 + 0.3708\zeta^3 \\ &\quad \text{then the number of humps} = 1; \\ &\text{if } I < 0.8737 + 1.0616\zeta - 0.2847\zeta^2 + 3.2461\zeta^3 \\ &\quad \text{then the number of humps} = 2, \end{aligned} \tag{31}$$

SI Shapers with more than two humps in their sensitivity curve can be determined, but they are rarely needed, so their development will be left out of this presentation.

Once the number of sensitivity curve humps has been determined, we must then determine how many impulses will be necessary to achieve the desired insensitivity. The number of impulses for $V=0.05$ can be determined from the following decision tree:

$$\begin{aligned} &\text{if } I < 0.06363 + 0.01044 \zeta + 0.07064 \zeta^2 + 0.40815 \zeta^3, \\ &\quad \text{then the number of impulses} = 2; \\ &\text{if } I < 0.3991 + 0.6313\zeta + 0.3559\zeta^2 + 2.3052\zeta^3 \\ &\text{and } I > 0.06363 + 0.01044\zeta + 0.07064\zeta^2 + 0.40815\zeta^3, \\ &\quad \text{then the number of impulses} = 3; \\ &\text{if } I > 0.3991 + 0.6313\zeta + 0.3559\zeta^2 + 2.3052\zeta^3 \\ &\quad \text{then the number of impulses} = 4. \end{aligned} \tag{32}$$

Once the number of sensitivity curve humps and the number of impulses have been

determined, the equations to be solved to develop the shaper can be stated explicitly. The constraint equations can be divided into three groups. First, there are two equations which set the limits of the sensitivity range regardless of the number of sensitivity curve humps:

$$V = e^{-\zeta \omega_{l0} t_d} \left\{ (\sum A_i e^{\zeta \omega_{l0} t_i} \cos(\omega_{l0} \sqrt{1-\zeta^2} t_i))^2 + (\sum A_i e^{\zeta \omega_{l0} t_i} \sin(\omega_{l0} \sqrt{1-\zeta^2} t_i))^2 \right\}^{1/2} \quad (33)$$

$$V = e^{-\zeta \omega_{hi} t_d} \left\{ (\sum A_i e^{\zeta \omega_{hi} t_i} \cos(\omega_{hi} \sqrt{1-\zeta^2} t_i))^2 + (\sum A_i e^{\zeta \omega_{hi} t_i} \sin(\omega_{hi} \sqrt{1-\zeta^2} t_i))^2 \right\}^{1/2}$$

where ω_{hi} and ω_{l0} are the two frequencies which specify the bounds on the desired insensitivity range.

Second, the vibration and derivative of the vibration curve must be fixed at each hump of the sensitivity curve. These equations are:

$$V = e^{-\zeta \omega_i t_n} \sqrt{\left(\sum_{i=1}^n A_i e^{\zeta \omega_i t_i} \cos(\omega_i \sqrt{1-\zeta^2} t_i) \right)^2 + \left(\sum_{i=1}^n A_i e^{\zeta \omega_i t_i} \sin(\omega_i \sqrt{1-\zeta^2} t_i) \right)^2}$$

$$0 = \frac{d}{d\omega_i} \left(e^{-\zeta \omega_i t_n} \sqrt{\left(\sum_{i=1}^n A_i e^{\zeta \omega_i t_i} \cos(\omega_i \sqrt{1-\zeta^2} t_i) \right)^2 + \left(\sum_{i=1}^n A_i e^{\zeta \omega_i t_i} \sin(\omega_i \sqrt{1-\zeta^2} t_i) \right)^2} \right) \quad (34)$$

where ω_i are the unknown frequencies at which the humps occur.

Finally, the sensitivity curve must be forced to zero between each hump. This gives rise to the equations:

$$0 = e^{-\zeta \omega z_j t_n} \sqrt{\left(\sum_{i=1}^n A_i e^{\zeta \omega z_j t_i} \cos(\omega z_j \sqrt{1-\zeta^2} t_i) \right)^2 + \left(\sum_{i=1}^n A_i e^{\zeta \omega z_j t_i} \sin(\omega z_j \sqrt{1-\zeta^2} t_i) \right)^2} \quad (35)$$

where ωz_j are unknown frequencies that lie interlaced between the ω_i hump frequencies. For example, $\omega z_1 < \omega_1 < \omega z_2 < \omega_2 \dots$

We have followed this process and determined the SI shaper for a wide range of I , ζ , and V . For example, the SI shaper for $V=0.05$, number of humps = 0, and number of impulses = 2, is described by:

$$\begin{aligned} A_1 &= 0.5 + 0.7623\zeta - 0.0320\zeta^2 - 0.2550\zeta^3 \\ A_2 &= 1 - A_1 \\ T_1 &= 0 \\ T_2 &= \left[\begin{array}{l} 0.4839 + 0.2522I + (-0.0253 - 0.0066I)\zeta \\ + (-0.02519 + 0.03407I)\zeta^2 + (-0.09932 - 0.0665I)\zeta^3 \end{array} \right] T_d \end{aligned} \quad (36)$$

The amplitudes and time locations of the impulses composing the SI shaper are complex functions of V , I , and ζ , as demonstrated by the need for decision trees. To graphically demonstrate the complexity, A_1 is plotted as a function of I and ζ in FIG. 12 for the case of $V=0.05$. FIG. 13 shows the corresponding curves for T_2 .

Negative Input Shapers

Introduction

Traditionally, the constraint equations used to determine input shapers have required positive values for the impulse amplitudes. However, move time can be significantly reduced by allowing the shaper to contain negative impulses. [Rappole, B. W.; Singer, N.C.; Seering, W.P. "Input Shaping™ with Negative Sequences for Reducing Vibrations in Flexible Structures," *Proceedings of 1993 Automatic Controls Conference*, San Francisco, CA]. considers the subject of time-optimal negative input shapers. Unfortunately, the method for obtaining time-optimal negative shapers presented in [Rappole, B. W.; Singer, N.C.; Seering, W.P. "Input Shaping™ with Negative Sequences for Reducing Vibrations in Flexible Structures," *Proceedings of 1993 Automatic Controls Conference*, San Francisco, CA] required the numerical solution of a set of simultaneous transcendental equations. This invention presents a look-up method that allows the design of negative input shapers without solving a set of complicated equations.

The constraint equations used to design an input shaper can vary greatly depending on the application, but they always include limitations on the amplitude of vibration at problematic frequencies. The constraint on vibration amplitude can be expressed as the ratio of residual vibration amplitude with shaping to that without shaping. This *percentage vibration* ratio is given by:

$$\%Vibration = e^{-\zeta\omega t_n} \{ (\sum A_i e^{\zeta\omega t_i} \cos(\omega\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\omega t_i} \sin(\omega\sqrt{1-\zeta^2}t_i))^2 \}^{1/2} \quad (1)$$

where A_i and t_i are the amplitudes and time locations of the impulses, t_n is the time of the last impulse, ω is the vibration frequency, and ζ is the damping ratio.

In addition to limiting vibration amplitude, most shaping methods require some amount of insensitivity to modeling errors. A shaper's insensitivity is displayed by a *sensitivity curve*: a plot of vibration versus frequency, (Eq. 1 plotted as a function of ω). A sensitivity curve reveals how much residual vibration will exist when there is an error in the estimation of the vibration frequency.

Most of the single-mode input shapers discussed in the literature have a length equal to one period of the vibration. This section describes three types of negative input shapers, each much shorter than one period. They satisfy the following three types of constraints:

- ZV (Zero Vibration at a specific frequency)[Singer, N.; Seering, W. "Preshaping Command Inputs to Reduce System Vibration," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 112, No. 1, pp. 76-82, March, 1990, Smith, O.J.M. *Feedback Control Systems*. pgs. 331-347, McGraw-Hill Book Company, Inc., New York, 1958].

- ZVD (Zero Vibration and zero Derivative of Eq. (1) at the modeling frequency)[Singer, N.; Seering, W. "Preshaping Command Inputs to Reduce System Vibration," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 112, No. 1, pp. 76-82, March, 1990].
- EI (Extra-Insensitive – a small level of vibration at the modeling frequency is allowed and the insensitivity is maximized)[Singhose, W.; Seering, W.; Singer, N. "Residual Vibration Reduction Using Vector Diagrams to Generate Shaped Inputs," *ASME Journal of Mechanical Design*, June 1994.].

For most of the constraints, a closed-form solution cannot be derived. However, we obtained numerical solutions using GAMS[Brooke, Kendrick, and Meeraus, *GAMS: A User's Guide*, Redwood City, CA, The Scientific Press, 1988], a linear and non-linear programming package. We will present tables that allow the public to design a negative input shaper without resorting to linear or non-linear programming. A later section presents methods for dealing with the high-mode excitation that may occur when negative input shapers are used.

Over-Currenting with Negative Shapers

Unlike shapers containing only positive impulses, negative shapers can lead to shaped command profiles which exceed the magnitude of the unshaped command for small periods of time. These small periods of over-currenting are not a problem for most applications because amplifiers and motors have peak current capabilities much larger than allowable steady state levels.

We can control the amount of over-currenting by limiting the partial sums of the impulse sequence to below a peak level, P. For example, a negative shaper with impulse amplitudes of A_1 , A_2 , and A_3 can be limited by the constraints:

$$A_1 \leq P, \quad A_1 + A_2 \leq P, \quad A_1 + A_2 + A_3 \leq P \quad (2)$$

When the constraints of Eq. (2) are enforced, almost the entire shaped command will be within $\pm P_{Max}$, where Max is the maximum unshaped command level. There will, however, still be brief periods when the shaped input exceeds P_{Max} , as illustrated in FIGS. 14a, 14b and 14c. FIG. 14a graphs the unshaped input, FIG. 14b depicts a typical negative shaper designed with $P=1$, that is, the amplitudes are $A_i=[1, -2, 1]$, and FIG. 14c graphs the resulting shaped input. The unshaped command is the acceleration associated with a trapezoidal velocity profile.

The amount of time the shaped command requires over-currenting is a function of the acceleration limit, velocity limit, move distance, system frequency, and input shaper. Numerous shaped commands were generated while varying the above parameters. For all

reasonable moves with $P=1$, only 1-3% of the shaped input required over-currenting. For $P > 1$, the amount of over-currenting increases with P .

A controls engineer who wants to design a negative input shaper cannot go wrong by choosing $P=1$. Almost any system can handle the brief periods of over-currenting. As we will see, a negative shaper designed with $P=1$ will move a system considerably faster than an all positive shaper. If even faster response time is required, P can be increased, but the duty cycle of the amplifiers and motors must be considered.

Physical systems can tolerate peak currents for only small durations. The amount of time that the system can withstand peak currents is usually determined by thermal considerations. This specification is usually referred to as "duty cycle". For example, a system might have a continuous current of 10 amps but tolerate a 50% duty cycle at 20 amps. Negative input shapers can be designed by raising the value of P until the duty cycle limit is reached.

Negative ZV (Zero Vibration) Shapers

The above constraints on the partial sum of the impulses in a shaper must be combined with constraints on the residual vibration. ZV constraints only require zero residual vibration at the frequencies of interest. Because there is no insensitivity to modeling errors, ZV shapers will not work well for most applications. We present them here because they are the shortest and, therefore, the highest performance shapers when the system frequencies are known very accurately.

When the ZV constraints are satisfied with a minimized shaper length, the solution converges to a three-impulse shaper with amplitudes of: $[P, -2P, P+1]$ for all values of damping ratio, ζ , and peak partial sum, P . The impulse amplitudes are easily described, however, the time locations of the impulses are rather complex functions of ζ and P . The impulse time locations also depend on the period of vibration, T , but the dependence is trivial. The time locations scale linearly with T .

When $\zeta=0$, the problem simplifies, and we can derive an analytic solution for the negative ZV shaper. From the shaper length minimization constraint, we know:

$$\begin{aligned} A_1 &= P \\ A_2 &= -2P \\ A_3 &= P+1 \end{aligned} \tag{3}$$

When we enforce the zero vibration constraint by setting Eq. (1) equal to zero, we get two constraint equations because the sin and cos terms are squared and must, therefore, both be zero. The resulting equations are:

$$-2P\sin(\omega t_2) + (P+1)\sin(\omega t_3) = 0 \tag{4}$$

$$P - 2P\cos(\omega t_2) + (P+1)\cos(\omega t_3) = 0 \quad (5)$$

t_1 does not appear in the constraint equations because it must be zero to achieve the minimal shaper length. Equations 3 and 4 can be solved for both t_2 and t_3 . The solutions are:

$$t_2 = \frac{T}{2\pi} \cos^{-1}\left(\frac{4P^2 - 2P - 1}{4P^2}\right) \quad (6)$$

$$t_3 = \frac{T}{2\pi} \cos^{-1}\left(\frac{2P^2 - 2P - 1}{2P(P+1)}\right)$$

Equations 3 and 6 describe the negative ZV shaper for undamped systems. The length of the negative shaper is 0.29T when $P=1$, as compared with 0.5T for the positive ZV shaper. Eq. (6) reveals the length of the negative ZV shaper decreases as P increases. When P is increased from 1 to 3, the shaper is shortened 0.116T, which is considerably greater than the decrease of 0.037T that occurs when P is further increased from 3 to 5. There is a decreasing return in time savings when P is increased further. Additionally, too high a value for P will lead to saturation of the actuators, as mentioned previously.

A negative shaper will perform slightly worse than a positive shaper in the presence of modeling errors, even though they are both derived with the same performance constraints. To quantify this effect, we define a numerical value for a shaper's sensitivity to modeling errors. *Insensitivity* is the width of the sensitivity curve at a given level of vibration. Vibration levels of 5% and 10% are commonly used to calculate insensitivity. For example, the positive ZV shaper has a 5% insensitivity of 0.065; that is, the percentage vibration is less than 5% from 0.9675ω to 1.0325ω , ($1.0325 - 0.9675 = 0.065$). The corresponding negative sequence has a 5% insensitivity of 0.055. This result is portrayed in FIG. 15. The normalized frequency axis in FIG. 15 is ω/ω_m , where the actual frequency of vibration is ω , and ω_m is the modeling frequency.

The time savings gained by using negative shapers comes with the risk of high-mode excitation. To assess this risk, we plot the shaper's sensitivity curve over a range of high frequencies. At frequencies where the sensitivity curve is above 100%, high-mode excitation can occur if the system has a second resonance. FIG. 16 compares the sensitivity curves for the positive ZV shaper and the negative ZV shapers for $P=1$ & 3. The positive shaper never exceeds 100%, but the negative shapers exceed this value over a large range of high frequencies.

For damped systems an analytic solution of the impulse times has not been found. However, curve fits to solutions obtained with GAMS were generated for $P=1, 2, \& 3$. The curve fits to t_2 and t_3 are shown in Table 1 along with an exact expression for t_3 that can be used instead of the curve fits once t_2 has been determined.

Table 1: Time Locations of Negative ZV Shaper

P = 1	P = 2	P = 3
$t_2 = (0.20963 + 0.22433\zeta)Td$	$t_2 = (0.12929+0.09393\zeta - 0.06204\zeta^2)Td$	$t_2 = (0.10089+0.05976\zeta - 0.05376\zeta^2)Td$
$t_3 = (0.29027+0.08865\zeta + 0.02646\zeta^2)Td$	$t_3 = (0.20975+0.02418\zeta - 0.07474\zeta^2)Td$	$t_3 = (0.17420+0.01145\zeta - 0.07317\zeta^2)Td$
Alternative to curve fits for all values of P:	$t_3 = \frac{\sqrt{1-\zeta^2}}{4\pi\zeta} \ln \left\{ \frac{P^2}{(P+1)^2} [1+4e^{2\zeta\omega_d t_2} \cdot 4e^{\zeta\omega_d t_2} \cos(\omega_d t_2)] \right\} Td$	

Negative ZVD (Zero Vibration & Derivative) Shapers

ZV shapers do not work well for most applications because they are sensitive to modeling errors, as shown in FIG. 15. To generate shapers that work on most real systems, we must add constraints that ensure insensitivity.

An often used insensitivity constraint proposed by Singer and Seering requires the derivative of the percentage vibration equation (Eq. 1) to be zero at the modeling frequency. To satisfy these ZVD constraints, the shaper must contain five impulses. If we minimize the sequence length, the amplitudes of the impulses are: $A_i=[P, -2P, 2P, -2P, P+1]$. The time location of each impulse is a complex function of ζ and P.

Curve fits to $t_2, t_3, t_4,$ and t_5 were obtained by holding P constant. Table 2 shows the curve fit description of the negative ZVD shapers for P = 1, 2, & 3.

Table 2: Time Locations of Negative ZVD Shapers

P = 1	P = 2	P = 3
$t_2 = (0.15236+0.23230\zeta + 0.09745\zeta^2)Td$	$t_2 = (0.11700+0.15424\zeta + 0.03449\zeta^2)Td$	$t_2 = (0.10022+0.11695\zeta + 0.00246\zeta^2)Td$
$t_3 = (0.27750+0.10237\zeta - 0.00612\zeta^2)Td$	$t_3 = (0.26041+0.11899\zeta - 0.05910\zeta^2)Td$	$t_3 = (0.24352+0.10877\zeta - 0.08790\zeta^2)Td$
$t_4 = (0.63139+0.33716\zeta - 0.07724\zeta^2)Td$	$t_4 = (0.49378+0.15092\zeta - 0.25380\zeta^2)Td$	$t_4 = (0.44109+0.11059\zeta - 0.23127\zeta^2)Td$
$t_5 = (0.67903+0.18179\zeta - 0.06008\zeta^2)Td$	$t_5 = (0.56273+0.04255\zeta - 0.19898\zeta^2)Td$	$t_5 = (0.51155+0.02121\zeta - 0.20054\zeta^2)Td$

FIG. 17 shows the ZVD shaper is substantially more insensitive than the ZV shaper. The 5% insensitivity of the negative ZVD shaper with P=1 is 0.253. This is a factor of 4.6 more than the negative ZV shaper.

The length of the negative ZVD shaper is only 68% of the positive ZVD input shaper when P=1. The time savings increases with P, as can be seen in FIG. 18. When P is increased from 1 to 3, the sequence is shortened 0.167Td, which is considerably larger than the decrease of 0.057Td that occurs when P is further increased from 3 to 5.

Negative EI (Extra-Insensitive) Shapers

As an alternative to ZV or ZVD constraints, we can achieve significantly more insensitivity by relaxing the constraint of zero vibration at the damped modeling frequency, ω_{dm} . If we limit the residual vibration at the modeling frequency to some small value, V, instead of

zero, we can enforce the zero vibration constraint at two frequencies, one higher than ωd_m and the other lower than ωd_m . This set of constraints leads to input shapers that are essentially the same length in time as the ZVD shapers, but have more insensitivity. The constraints, in equation form, are:

$$V = e^{-\zeta \alpha x} \left\{ (\sum A_i e^{\zeta \alpha x_i} \cos(\omega \sqrt{1-\zeta^2} t_i))^2 + (\sum A_i e^{\zeta \alpha x_i} \sin(\omega \sqrt{1-\zeta^2} t_i))^2 \right\}^{1/2} \quad (7)$$

$$\sum A_i e^{\zeta \omega t_i} \sin(t_i \omega d_{hi}) = \sum A_i e^{\zeta \omega t_i} \sin(t_i \omega d_{low}) = 0 \quad (8)$$

$$\sum A_i e^{\zeta \omega t_i} \cos(t_i \omega d_{hi}) = \sum A_i e^{\zeta \omega t_i} \cos(t_i \omega d_{low}) = 0 \quad (9)$$

where, ωd_{hi} and ωd_{low} are the frequencies where the sensitivity curve is forced to zero. Eq. 7 contains a t_5 term because the shaper contains five impulses.

Using GAMS, we solved the EI constraints over a suitable range of V , ζ , and P . The shaper's amplitudes are: $[P, -2P, 2P, -2P, P+1]$, just as in the case of ZVD constraints. Table 3 shows curve fits to the time locations of the negative EI shapers for $P=1, 2, \& 3$ when $V = 5\%$. Table 4 is the same information for the case of $V = 10\%$.

Table 3: Time Locations of V=5% Negative EI Shaper

P = 1	P = 2	P = 3
$t_2 = (0.15687+0.24004\zeta + 0.20367\zeta^2)Td$	$t_2 = (0.11955+0.16127\zeta + 0.05206\zeta^2)Td$	$t_2 = (0.10219+0.12192\zeta + 0.01197\zeta^2)Td$
$t_3 = (0.28151+0.10650\zeta + 0.09280\zeta^2)Td$	$t_3 = (0.26356+0.12551\zeta - 0.03963\zeta^2)Td$	$t_3 = (0.24639+0.11404\zeta - 0.07655\zeta^2)Td$
$t_4 = (0.63431+0.33886\zeta - 0.12776\zeta^2)Td$	$t_4 = (0.49804+0.15508\zeta - 0.24101\zeta^2)Td$	$t_4 = (0.44526+0.11468\zeta - 0.22230\zeta^2)Td$
$t_5 = (0.68414+0.18236\zeta + 0.00839\zeta^2)Td$	$t_5 = (0.56866+0.04558\zeta - 0.18732\zeta^2)Td$	$t_5 = (0.51719+0.02439\zeta - 0.19225\zeta^2)Td$

Table 4: Time Locations of V=10% Negative EI Shaper

P = 1	P = 2	P = 3
$t_2 = (0.16136+0.24772\zeta + 0.31367\zeta^2)Td$	$t_2 = (0.12207+0.16808\zeta + 0.07038\zeta^2)Td$	$t_2 = (0.10412+0.12667\zeta + 0.02201\zeta^2)Td$
$t_3 = (0.28547+0.11044\zeta + 0.19967\zeta^2)Td$	$t_3 = (0.26661+0.13190\zeta - 0.019712\zeta^2)Td$	$t_3 = (0.24916+0.11908\zeta - 0.06480\zeta^2)Td$
$t_4 = (0.63719+0.33687\zeta - 0.14612\zeta^2)Td$	$t_4 = (0.50210+0.15873\zeta - 0.22743\zeta^2)Td$	$t_4 = (0.44925+0.11835\zeta - 0.21300\zeta^2)Td$
$t_5 = (0.68919+0.17941\zeta - 0.01215\zeta^2)Td$	$t_5 = (0.57439+0.04813\zeta - 0.17499\zeta^2)Td$	$t_5 = (0.52261+0.02720\zeta - 0.18377\zeta^2)Td$

By examining Tables 2-4, we find that the EI shapers are essentially the same length as the ZVD shapers regardless of the values for V , ζ , or P . However, FIG. 19 clearly shows that the EI shapers are more insensitive to modeling errors than the ZVD shapers. For the case shown in FIG. 19 ($V=5\%$), the EI shaper has 40% more insensitivity than the ZVD shaper. (The EI shaper has a 5% insensitivity of 0.352, while the ZVD shaper has a 5% insensitivity of 0.252.)

The increase of insensitivity near the modeling frequency comes only from relaxing the zero vibration constraint at the modeling frequency. There are no additional costs

associated with the EI shapers. We have already seen that the length of the EI shaper is essentially the same as the ZVD, and FIG. 20 shows that there is little difference between the two at high frequencies.

When P is increased, some insensitivity around the modeling frequency is lost. If P is increased from 1 to 3, the EI shaper is shortened by 24%, while the 5% insensitivity drops from 0.352 to 0.333. The drop in insensitivity is very small given the large time savings.

While the example given above was for a five-impulse sequence with one hump in the sensitivity curve, the method may be extended to sequences with an arbitrary number of impulses and/or sensitivity curves with an arbitrary number of humps, in the same manner as discussed previously for extra-insensitive shapers.

Controlling High-Mode Excitation

As we have seen, negative shapers can increase vibration at unmodeled modes higher than the frequency for which they were designed. For high-mode excitation to occur, there must be a resonance at a frequency where the sensitivity curve exceeds 100%. Even if high-mode excitation occurs, Input Shaping™ will probably decrease the total amount of system vibration. This decrease results from the elimination of the low mode, which usually contains the majority of vibration amplitude.

In cases where high-mode excitation is performance-limiting, we have several options to chose from:

- 1) Give up the time savings gained by using a negative sequence and use a positive input shaper.
- 2) Add restrictions on the amplitudes of high-mode vibration and solve the augmented set of constraints.
- 3) Add a digital low-pass filter.
- 4) Use an input with no high-frequency content.

The first option is the easiest and most appropriate when increasing the speed of the system is not the highest priority. Option 2 is the highest performance solution because we can customize an input shaper to a specific system with a minimal time penalty. Unfortunately, the look-up method provided by Tables 1-4 has to be abandoned. Options 3&4 can still utilize Tables 1-4, however, there can be a large increase in the computational needs during run-time.

Option 2, the process of restricting only a few problematic high frequencies is best demonstrated with an example. Suppose we select a negative EI shaper to eliminate a 1 Hz mode from our system. However, when we use the shaper, we discover that a previously

unimportant mode at 7 Hz is excited more than in the unshaped case. We can eliminate this vibration by adding an equation to our set of constraints that limits the vibration at 7 Hz to 50% of the unshaped vibration. We could just as well use 30% or 60%; we just want to restrict the vibration to less than the unshaped level.

The time-optimal shaper meeting the above constraints was calculated with GAMS and is only 6% longer than the original EI shaper. Sensitivity curves for the EI shaper and the EI shaper with a 7 Hz vibration limitation are shown in FIG. 21. The vibration-limited shaper has the same number of impulses as the unconstrained shaper. However, the amplitudes are no longer: [1, -2, 2, -2, 2]. Instead, they are: [1, -2, 1.82, -1.82, 2].

If more than one high mode is problematic, we simply add a constraint equation for each mode of vibration and solve the augmented set of constraints. For each mode that is constrained, a small amount of time will be added to the shaper length. However, a negative shaper with high-mode constraints will continue to be shorter than a positive shaper for the low mode until constraints have been placed on a large number of high modes.

The technique of restricting the vibration at a few high modes is advantageous because:

- 1) The shaper is customized to a specific system and, therefore, it does not over-constrain the system.

- 2) The computational requirements during run-time are usually unaffected.

The drawback of this approach is that it requires the one-time solution of a set of simultaneous, transcendental constraint equations; Tables 1-4 cannot be used.

Instead of restricting a few high modes, we can eliminate all high frequencies by adding a low-pass filter. The low-pass filter is used in conjunction with a negative shaper of the designer's choice from Tables 1-4 to give a modified input shaper that will not excite modes in the filter's stop band. A time delay is added when we use a low-pass filter because the length of the modified shaper is equal to the length of the original shaper plus the length of the low-pass filter. The computational requirements can increase significantly because implementation of the modified shaper requires N more multiplies and adds than the original shaper, where N is the low-pass filter length number. Standard low-pass filters are often designed with an N value of 20-30, but N can range from 5 to values in the hundreds.

To implement the modified shaper, place the low-pass filter in series with the input shaper instead of convolving the shaper and filter together. This procedure reduces the computational load during run-time.

To demonstrate the technique of adding a low-pass filter, we will design a modified

shaper for the system discussed above. Once again, we will start with a negative EI shaper from Table 3 with $P=1$ to eliminate the 1 Hz mode. Next, for demonstration purposes, we add a low-pass filter designed with a pass band of 0-1 Hz, a stop band of 12-50 Hz, $N=13$, and a sampling frequency of 100 Hz. The high-mode sensitivity curve for the low-pass negative input shaper is shown in FIG. 22. The low-pass negative shaper has 18 impulses (5+13). This more than triples the computational requirement during run-time.

The main benefits of augmenting a negative input shaper with a low-pass filter are:

- 1) It requires no specific information about the high-mode frequencies, just a pass band and a stop band.
- 2) It eliminates a large range of high frequencies.
- 3) It uses well established filter design tools in combination with the solutions from Tables 1-4. This approach has two major drawbacks:

- 1) There is a longer time delay associated with this Input Shaping™ process than when we restrict just a few high modes. (The low-pass negative input shaper in our example is 19% longer than the original EI shaper.)

- 2) The run-time computation can be prohibitive.

We will not give an example of the fourth option for dealing with high-mode excitation, using an input function that does not contain energy at high frequencies. Instead, we will simply assert that the technique has more drawbacks than using a low-pass filter. It requires additional computation and performance will vary with the move distance.

Negative Shaper Experimental Results

Tests of the negative input shapers listed in Tables 1-4 were conducted on a rotary table. To ensure a vibratory response, a steel beam was mounted to the table surface, with 19 inches of the beam free to bend when the table was rotated. A 2 lb. mass was attached to the end of the beam to simulate a payload. The Inland Torque Motor used to rotate the table was equipped with an HP HEDS-6110 encoder with 44,000 counts per revolution. The motor was powered by an Aerotech DS16020 amplifier and the control signal was generated by a Macintosh Quadra 700 running a PD controller at 500 Hz. The controller generated trapezoidal velocity profiles based on maximum velocity and acceleration limits.

A one radian move was commanded and an FFT on the residual vibration revealed a mode near 2 Hz with a damping ratio of about 0.08. Negative ZV, ZVD, and EI shapers were then calculated from Tables 1-4. Input Shaping™ was enabled and the table response with each of the shapers was recorded. Additionally, the table response with a positive ZVD shaper was measured, so that the time savings from the negative shapers could be evaluated.

FIG. 23 compares the unshaped response of the table to the responses with the negative ZV, ZVD, and EI shapers. The negative ZV and EI shapers reduced the vibration to approximately 5% of the unshaped level, while the negative ZVD reduced the vibration to 0.8%. The ZV shaper left 5% of the vibration because it is very sensitive to modeling errors and our frequency identification has limited accuracy. The EI shaper left 5% of the vibration because it is designed to leave 5% vibration near the modeling frequency.

If we examine the 2% settling time, we find the response with the negative ZV shaper settled in 0.69 sec. and both the negative ZVD and EI responses settled in approximately 0.85 sec. The response with the positive ZVD shaper settled in 0.93 sec., while the unshaped response took 2.78 sec. to settle within 2% of the desired position.

If our evaluation was based only on the above results, we might choose the negative ZV shaper because it gives the fastest settling time. Or, we might choose the negative ZVD shaper because it gives the least amount of residual vibration. However, in most real systems, the actual frequency will deviate from the modeling frequency when the system geometry changes or a payload is picked up.

To evaluate the shapers in the presence of modeling errors, additional mass was added to the steel beam and the experiments were repeated. The extra mass resulted in a very large frequency shift of approximately 25%.

FIG. 24 compares the responses of the extra-mass system with the negative ZV, ZVD, and EI shapers. Also shown is the response to the positive ZVD shaper. In this case, the negative EI shaper is superior in both vibration suppression and settling time. Note that FIG. 24 is a close up of the residual vibration. The level of vibration with a 25% modeling error is still only about 20% of the unshaped vibration shown in FIG. 23. The unshaped vibration would be way off the scale if it were also plotted on FIG. 24. The experimental results are summarized in Table 5.

Table 5: Summary of Experimental Data

Shaper	Original System		Extra-Mass System (25% Lower Frequency)	
	Vibration (% of Unshaped)	2% Settling Time	Vibration (% of Unshaped)	2% Settling Time
None	100	2.78 sec.	122	> 4 sec.
Positive ZVD	0.6	0.93 sec.	21	1.18 sec.
Negative ZV	5.8	0.69 sec.	57	3.27 sec.
Negative ZVD	0.8	0.84 sec.	24	1.44 sec.
Negative EI (V=5%)	5.5	0.85 sec.	17	0.81 sec.

Extra-Insensitive Input Shapers for Controlling Flexible Spacecraft

Introduction

The problem of controlling flexible structures in the presence of modeling uncertainties and structural nonlinearities is an area of active research. The control techniques currently under investigation include: adding damping to the system, stiffening the structure, developing a sophisticated model and controller, and shaping the command signals. For space-based systems, the first two options can be sub-optimal because they usually require the addition of mass to the system. The third option is application specific and developments may be difficult to generalize. Generating command signals that do not excite unwanted dynamics is often the most appealing option.

This section will present a subset of the Input Shaping™ technique that is used for systems equipped with constant-magnitude actuators. This represents the case for most flexible spacecraft because reaction jets usually do not have variable force amplitude control, only on-off time control.

Some approximate, yet often effective, methods for extending Input Shaping™ to the case where only constant-magnitude inputs are used were presented in [Singhose, W. "A Vector Diagram Approach to Shaping Inputs for Vibration Reduction," *MIT Artificial Intelligence Lab Memo No. 1223*; March 1990]. Recently, Wie, Liu, and Sinha applied the patented method of [Singer, N.; Seering, W.; Pasch, K. "Shaping Command Inputs to Minimize Unwanted Dynamics", United States Patent 4,916,635, April 10, 1990] to the problem of controlling flexible spacecraft equipped with on-off reaction jets by combining Singer's constraint equations with a constant-magnitude constraint on the command [Liu, Q.; Wie, B. "Robust Time-Optimal Control of Uncertain Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, May-June 1992, Wie, B.; Sinha, R.; Liu, Q. "Robust Time-Optimal Control of Uncertain Structural Dynamic Systems," *Journal of Guidance, Control, and Dynamics*, Sep.-Oct. 1993]. They solved the set of constraint equations with a standard optimization package. The robust time-optimal control command they obtained consists of a bang-bang signal with six alternating pulses. While the solution obtained with this method works considerably better than an unshaped bang-bang controller, Singer's original robustness constraints are not optimal. The solution presented in [Liu, Q.; Wie, B. "Robust Time-Optimal Control of Uncertain Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, May-June 1992, Wie, B.; Sinha, R.; Liu, Q. "Robust Time-Optimal Control of Uncertain Structural Dynamic Systems," *Journal of Guidance, Control, and Dynamics*, Sep.-Oct. 1993] is not very robust because the constant-magnitude restriction is merely added to Singer's problem formulation without attempting to improve the robustness.

We will show how a more effective command signal can be generated by using extra-insensitive (EI) constraints similar to those presented in [Singhose, W.; Seering, W.;

Singer, N. "Residual Vibration Reduction Using Vector Diagrams to Generate Shaped Inputs," *ASME Journal of Mechanical Design*, June 1994]. The constraints are called extra-insensitive because they lead to input shapers that are significantly more insensitive to modeling errors and parameter variations than Singer's original robustness constraints. The set of constraint equations we develop is solved with GAMS [Brooke, Kendrick, and Meeraus, *GAMS: A User's Guide*, Redwood City, CA, The Scientific Press, 1988], a numerical optimization program.

To support our theoretical developments, we will present results from both computer simulations and hardware experiments.

Constraint Equation Derivation

One of the great advantages of the patented Input Shaping™ technique is that it requires only a very simple system model consisting of natural frequencies and damping ratios. Constraint equations based on the simple model are formulated so that the physically complex system responds with very little residual vibration. Additional equations can be added to the problem formulation ensuring that the residual vibration will remain small in the presence of modeling errors. The constraint equations can be written in a fairly simple form because the input is assumed to be a sequence of impulses and the response of a second order harmonic oscillator to an impulse is easily described. By solving the system of constraint equations, the amplitudes and time locations of the n impulses in the desired input shaper can be determined.

Because the present investigation is concerned with constant-magnitude force inputs, the system's mass (in addition to the natural frequency) must be considered if we are to move the system to a given position. For this reason, we base our development on the simple spring–mass model shown in FIG. 25. This model represents a system with a single flexible mode and a rigid body mode. A force input, $u(t)$, acts on mass m_1 , and is restricted to have a value of +1 or -1. In many space-based systems very small damping will exist, so we have left out a damping term to simplify development at this point.

For the system under consideration, the shaped input must consist only of positive and negative constant-magnitude force pulses attainable by jet-type actuators. To achieve this form of a command signal, the unshaped command signal must be a step in force, as shown in FIG. 26a. Furthermore, the input shaper must contain impulses with amplitudes of: $A_i = [1, -2, 2, \dots, -2, 1]$, where $i = 1, \dots, n$, and n is odd, as in FIG. 26b. This will ensure the shaped input is a bang-bang series of alternating sign, constant-magnitude pulses of variable width, as shown in FIG. 26c. This restriction on the impulse amplitudes represents our first constraint equations, and can be written as:

$$\begin{aligned}
A_j &= 1 & j &= 1, n \\
A_i &= 2(-1)^{i-1} & i &= 2, \dots, n-1
\end{aligned} \tag{1}$$

The next constraint equation we will present ensures that the center of mass of the system will move the desired distance. To determine the movement of the center of mass, we assume an infinite spring stiffness with the motion described as:

$$\ddot{x}_2 = \frac{u(t)}{(m_1 + m_2)} \tag{2}$$

By integrating Eq. 2 twice we get the desired constraint equation:

$$x_d = \iint \frac{u(t)}{(m_1 + m_2)} dt \tag{3}$$

where x_d is the desired move distance. At first glance Eq. 3 might look formidable, especially because we need to solve it simultaneously with several other equations, but $u(t)$ has a very simple form – it is either +1 or -1. Note that Eq. 3 must be used whether or not Input Shaping™ is used.

Eq. 3 ensures that the rigid body motion will be as desired, but we also need to limit the amount of residual vibration that occurs when the system reaches its desired setpoint. The constraint on vibration amplitude can be expressed as a ratio of residual vibration amplitude with Input Shaping™ divided by residual vibration amplitude without shaping. This percentage vibration is given by:

$$V = e^{-\zeta\omega t_n} \{(\sum A_i e^{\zeta\omega t_i} \sin(\omega\sqrt{1-\zeta^2} t_i))^2 + (\sum A_i e^{\zeta\omega t_i} \cos(\omega\sqrt{1-\zeta^2} t_i))^2\}^{1/2} \tag{4}$$

where A_i and t_i are the amplitudes and time locations of the impulses, t_n is the time of the last impulse, ω is the undamped natural frequency, and ζ is the damping ratio. The simple model in FIG. 25 does not include a damper, so Eq. 4 can be reduced considerably by setting $\zeta=0$.

By including Eq. 4 in our set of constraint equations, we can guarantee that there will be only the amount of residual vibration specified by the quantity V when the system's frequency is exactly ω and its damping ratio is exactly ζ . However, if the frequency is slightly different than ω , then there may exist a large amount of residual vibration. To make an input shaper robust to parameter variations, we can differentiate Eq. 4 with respect to ω and set the result equal to zero. In equation form, we get:

$$0 = \frac{d}{d\omega} \left(e^{-\zeta\omega t_n} \{(\sum A_i e^{\zeta\omega t_i} \sin(\omega\sqrt{1-\zeta^2} t_i))^2 + (\sum A_i e^{\zeta\omega t_i} \cos(\omega\sqrt{1-\zeta^2} t_i))^2\}^{1/2} \right) \tag{5}$$

This constraint ensures that the amount of residual vibration will not vary greatly from the quantity V when the frequency varies somewhat from ω .

The final constraint we will introduce ensures that our solution is time-optimal, that is, it is the shortest command signal that meets the other constraints. The constraint is simply:

minimize(t_n) (6)

where t_n is the time of the last impulse.

Eqs. 1 and 3-6 form the set of constraint equations that we will use to design input shapers that generate constant-magnitude pulse signals. Different solutions will be obtained by solving these constraints in various combinations.

Input Shaper Comparison

In this section we present several different input shapers that can be derived using the above constraint equations. The first two shapers have been presented elsewhere in the literature, but we will briefly describe them here so that we can use them to evaluate the performance of the EI shapers.

The first input shaper we will discuss is the one resulting from the solution of Eqs. 1,3, 4 and 6 with $n=5$ and $V = 0$. Notice we have left out the constraint on the derivative of the residual vibration, Eq. 5. This solution is significant because it is the shortest input shaper that meets the constraint of zero residual vibration. Wie and Liu refer to this shaper as the "Flexible-Body Time-Optimal" control in [Liu, Q.; Wie, B. "Robust Time-Optimal Control of Uncertain Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, May-June 1992], but we will call it the zero-vibration (ZV) input shaper because it meets the constraint of zero residual vibration when the system model is exact. The impulse sequence composing this shaper for $m_1=m_2=k=x_d=1$ is given by Liu and Wie as

$$\begin{bmatrix} A_i \\ T_i \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & -2 & 1 \\ 0 & 1.00268 & 2.10893 & 3.21518 & 4.21786 \end{bmatrix}$$

where, A_i is the amplitude of the i th impulse and T_i is the time location of the i th impulse in seconds. Wie and Liu note that perturbations in the spring constant, k , result in significant residual vibrations of the flexible mode. Changes in the mass values would, of course, also lead to residual vibration. A sensitivity curve, a plot of residual vibration (Eq. 4) as a function of ω , reveals how variations in both mass and spring values effect the level of residual vibration. The solid line of FIG. 27 shows the sensitivity curve for the ZV input shaper. Notice that at the nominal frequency of 1.41 rad = 0.225 Hz, which occurs when our model is exact, there is zero residual vibration. However, the amount of residual vibration increases rapidly as the frequency deviates from the nominal frequency.

We can obtain a measure of a shaper's performance in the presence of modeling errors by defining a shaper's *insensitivity* as the width of the sensitivity curve at a given level of vibration. Vibration levels of 5% and 10% are commonly used to calculate insensitivity. The 5% insensitivity for the ZV shaper is 0.0191; that is, the residual vibration is less than 5% of the unshaped vibration from 0.9903 ω to 1.0094 ω , (1.0094 -

0.9903 = 0.0191). To allow comparison, the insensitivity is normalized by the nominal frequency.

The second input shaper we will discuss is derived by adding Singer's original insensitivity constraint (Eq. 5) to the set of equations used to solve for the ZV input shaper. This shaper was labeled the "Robust Time-Optimal" control in [Liu, Q.; Wie, B. "Robust Time-Optimal Control of Uncertain Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, May-June 1992]. However, we will call it the zero-vibration and derivative (ZVD) shaper because it meets the requirements of both zero residual vibration (Eq. 4) and zero derivative of the residual vibration expression (Eq. 5), when the system model is exact. To achieve a solution to this set of constraints, n must be increased from 5 to 7. The impulse sequence composing this shaper is:

$$\begin{bmatrix} A_i \\ T_i \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & -2 & 2 & -2 & 1 \\ 0 & 0.7124 & 1.6563 & 2.9330 & 4.2097 & 5.1536 & 5.8660 \end{bmatrix}$$

The dotted-dashed line of FIG. 27 graphs the sensitivity curves for the ZVD input shapers. The superiority of the ZVD shaper compared to the ZV shaper in terms of insensitivity to modeling errors is obvious. If we compare its 5% insensitivity of 0.1505 (1.0700ω to 0.9195ω) to the 0.0191 insensitivity of the ZV shaper, we can say it is 7.9 times more insensitive to modeling errors or parameter variations.

It should be noted that the length of the command signal must be increased from 4.218 sec. to 5.866 sec. to achieve the added insensitivity to frequency changes, a small cost for the given benefit in most cases.

The trade-off between insensitivity and the maneuver time continually arises in the design of input shapers. We will now present a technique for deriving a shaper that is even more insensitive to parameter variations than the ZVD shaper, but requires essentially no additional time. This extra-insensitive input shaper is determined by setting V to a non-zero value in Eq. 4 when ω is equal to the nominal frequency. We then set Eq. 4 equal to zero at two other undetermined frequencies, one higher than the nominal frequency, ω_{hi} , and the other lower, ω_{low} . The complete set of constraint equations will be listed here for clarity:

$$\begin{aligned} A_j &= 1 & j &= 1, n \\ A_i &= 2(-1)^{i-1} & i &= 2, \dots, n-1 \end{aligned} \tag{7}$$

$$x_d = \iint \frac{u(t)}{(m_1 + m_2)} dt \tag{8}$$

$$V = e^{-\zeta\alpha n} \{ (\sum A_i e^{\zeta\alpha i} \sin(\omega\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\alpha i} \cos(\omega\sqrt{1-\zeta^2}t_i))^2 \}^{1/2} \tag{9}$$

$$0 = e^{-\zeta\omega_{hi}n} \{ (\sum A_i e^{\zeta\omega_{hi}i} \sin(\omega_{hi}\sqrt{1-\zeta^2}t_i))^2 + (\sum A_i e^{\zeta\omega_{hi}i} \cos(\omega_{hi}\sqrt{1-\zeta^2}t_i))^2 \}^{1/2} \tag{10}$$

$$0 = e^{-\zeta \omega_{low} t_n} \left\{ (\sum A_i e^{\zeta \omega_{low} t_i} \sin(\omega_{low} \sqrt{1-\zeta^2} t_i))^2 + (\sum A_i e^{\zeta \omega_{low} t_i} \cos(\omega_{low} \sqrt{1-\zeta^2} t_i))^2 \right\}^{1/2} \quad (11)$$

$$0 = \frac{d}{d\omega} \left(e^{-\zeta \omega t_n} \left\{ (\sum A_i e^{\zeta \omega t_i} \sin(\omega \sqrt{1-\zeta^2} t_i))^2 + (\sum A_i e^{\zeta \omega t_i} \cos(\omega \sqrt{1-\zeta^2} t_i))^2 \right\}^{1/2} \right) \quad (12)$$

$$\text{minimize } (t_n) \quad (13)$$

Eqs. 7-13 were solved using GAMS, a numerical optimization tool, for the case of $n=7$, $V=0.05=5\%$ and $m_1=m_2=k=x_d=1$. The impulse sequence composing this extra-insensitive shaper is:

$$\begin{bmatrix} A_i \\ T_i \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & -2 & 2 & -2 & 1 \\ 0 & 0.7188 & 1.6731 & 2.9465 & 4.2199 & 5.1742 & 5.8930 \end{bmatrix}$$

This EI shaper is compared to the two previous input shapers in FIG. 27. It has a 5% insensitivity of 0.2118 (1.0957 ω to 0.8839 ω). This is 1.4 times more insensitive than the ZVD input shaper and 11.1 times more insensitive than the ZV input shaper. The significant increase in insensitivity that comes with the EI shaper is obtained with an almost negligible increase in command signal length (5.893 sec. as compared to 5.866 sec. for the ZVD shaper).

It is interesting to compare the insensitivity for the above shapers, which lead to constant-magnitude pulse commands, to the more general input shapers that allow for variable-magnitude commands. For the case of the input shaper derived with only the zero-vibration constraints, we have already stated the 5% insensitivity is 0.0191. This is considerably less than the insensitivity of 0.065 that is obtained with the variable-magnitude input shaper derived with the zero-vibration constraints. Table 6 displays this insensitivity comparison and the corresponding comparisons for the ZVD shaper and the EI shaper. Table 6 clearly shows that the constraint requiring constant-magnitude commands leads to large decreases in insensitivity. For this reason, the more general input shapers should be used if the actual system commands are not restricted to constant-magnitude pulses.

Table 6: The Reduction in Insensitivity Caused By the Constant-Magnitude Constraint

Shaper Type	Variable-Magnitude Commands	Constant-Magnitude Pulses	Reduction in Insensitivity
ZV	0.065	0.0191	71%
ZVD	0.287	0.1505	48%
EI	0.399	0.2118	47%

Input Shaper Verification

A computer simulation of the system shown in FIG. 25 was constructed in MATLAB. The simulation was performed using the command signal generated with the ZVD input shaper and the $V=5\%$ EI input shaper. FIG. 28a graphs the system response to the ZVD shaper and FIG. 28b graphs the response to the EI shaper when k is varied from the nominal value of 1.0 to 0.7 in steps of -0.1. The envelope on the residual vibration is 32% lower for the EI shaper than for the ZVD shaper. For almost any variation of k , the envelope will be smaller for the EI shaper.

The sensitivity curves in FIG. 27 can be interpreted as giving the relative size of the residual vibration amplitude envelope as a function of ω . The only time the ZVD shaper outperforms the EI shaper is when the frequency is in the range of 0.945ω to 1.050ω . We should remember, however, that Input Shaping™ is used because there is uncertainty in the system model or the frequency is expected to change during operation. If it is known that the system will operate within $\pm 5\%$ of its nominal frequency, then a robust input shaper is not needed. In physical systems this is almost never the case, therefore, an EI shaper is almost always preferred to a ZVD shaper.

As a further test of the EI input shaper for constant-magnitude pulse systems, we conducted hardware tests on a rotary table. To ensure a vibratory response, a 24 in. steel beam was mounted to the table surface and a 2 lb. mass was attached to the end of the beam. The Inland Torque Motor used to rotate the table was equipped with an HP HEDS-6110 encoder with 44,000 counts per revolution. The motor was powered by an Aerotech DS16020 amplifier and the control signal was generated by a Motion Engineering PC/DSP Motion Controller. A bang-bang acceleration command was given to the table and an FFT was performed on the residual vibration. The frequency spectrum of FIG. 29 shows that the table's dominant low mode is approximately 2 Hz, and its second mode approximately 8 Hz.

A $V=5\%$, EI constant-magnitude shaper was designed for the 2 Hz mode and a 1 radian move. FIG. 30 compares the shaped response and the response to an unshaped bang-bang command that also moves the system 1 radian. The EI shaper reduced the vibration to about 7% of the unshaped vibration level. This is very close to the theoretical reduction of 5%. The small deviation from theory can be explained by errors in our choice of modeling frequency and damping ratio. Additionally, our assumption that the rotary table is a single mode system is not entirely accurate because a second mode at 8 Hz is clearly visible in the frequency spectrum.

Multiple-Mode Impulse Shaping Sequences

Introduction

Singer and Seering [Singer, N., and Seering, W, "Preshaping Command Inputs to Reduce System Vibration," *Journal of Dynamic Systems, Measurement, and Control*, March, 1990] demonstrate that sequences for systems with multiple modes of vibration can be generated by convolving together single-mode sequences for each of the frequencies. This method of shaping for m modes will result in $3m$ impulses in the sequence, and the sequence will extend the theoretical move time by an amount equal to the sum of the periods of vibration of the modes being shaped. In an effort to reduce the number of impulses and decrease the sequence duration, Hyde and Seering [Hyde, J. and Seering, W., "Using Input Command Pre-Shaping to Suppress Multiple Mode Vibration," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA: April, 1991] developed a method for generating multiple-mode sequences that simultaneously solves the constraint equations for all modes. Direct solution of the constraints for multiple-modes yields sequences with only $2m + 1$ impulses. Hyde's method generates sequences that produce good vibration reduction characteristics, while being generally shorter than convolved sequences. Unfortunately, this solution method requires complex optimization routines to generate the sequences.

Observation shows that many of the sequences generated by Hyde's method are symmetric about the middle impulse. The undamped systems have sequences that are symmetric in both time and amplitude, while the systems with damping have sequences that are symmetric only in time. We use this symmetry to derive an analytic solution for a two-mode, undamped system.

Two-Mode Sequences

The method of command shaping was developed from a second-order, linear model of a system with natural frequency, ω , and damping ratio, ζ [Singer, Neil C., *Residual Vibration Reduction in Computer Controlled Machines*, Massachusetts Institute of Technology Artificial Intelligence Laboratory Technical Report No. 1030, 1989] . This system has a known response, $y(t)$, to a sequence of impulses of amplitudes, A_i , applied at times, t_i .

$$y(t) = \sum_{i=1}^n \frac{A_i \omega}{\sqrt{1-\zeta^2}} e^{-\zeta\omega(t-t_i)} \sin[(t-t_i)\omega\sqrt{1-\zeta^2}] \quad (1)$$

Singer generates his sequences using the following constraints derived from (1). We

present the derivation here because the multiple-mode sequences are derived from the same equations. The amplitude of the response to a series of n impulses is constrained to zero after the sequence of impulses:

$$\sum_{i=1}^n A_i e^{-\zeta\omega(t-t_i)} \sin(t_i \omega \sqrt{1-\zeta^2}) = 0 \quad (2)$$

$$\sum_{i=1}^n A_i e^{-\zeta\omega(t-t_i)} \cos(t_i \omega \sqrt{1-\zeta^2}) = 0 \quad (3)$$

The derivatives of (2) and (3) with respect to natural frequency are also constrained to zero, adding insensitivity to modeling errors:

$$\sum_{i=1}^n A_i t_i e^{-\zeta\omega(t-t_i)} \sin(t_i \omega \sqrt{1-\zeta^2}) = 0 \quad (4)$$

$$\sum_{i=1}^n A_i t_i e^{-\zeta\omega(t-t_i)} \cos(t_i \omega \sqrt{1-\zeta^2}) = 0 \quad (5)$$

Three other constraints are specified:

$$t_1 = 0 \quad (6)$$

$$\sum_{i=1}^n A_i = 1 \quad (7)$$

$$A_i \geq 0 \quad (8)$$

Eqn. (6) sets the time origin for the shaper, eqn. (7) guarantees unity gain, and eqn. (8) constrains the amplitudes to be positive, insuring that higher frequencies will not be excited by the shaper. With these seven constraints, (2) - (8), the *three-impulse sequence* is generated.

The two-mode sequence requires a different set of constraints: (2) - (5) are repeated for the ω and ζ of each of the two frequencies and (6) and (7) are included. These ten constraint equations yield a five-impulse sequence for two modes of vibration. When ζ is zero for both modes, a closed-form solution can be found. We assume that a five-impulse, symmetric solution exists to equations (2) - (8) for two modes, with amplitudes A_i and times t_i , substitute

$$T_i = t_i - t_3 \quad (9)$$

into (1), and re-derive (2) - (5) in terms of the T_i , with $\zeta = 0.0$. The symmetry assumption makes $T_1 = -T_5$, $T_2 = -T_4$, $A_1 = A_5$, and $A_2 = A_4$. This derivation yields four equations (10) & (11), which with (6) and (7), define a five-impulse, symmetric sequence for two frequencies of vibration, ω_j :

$$\frac{1}{4} = A_1 \sin^2\left(\frac{\omega_j T_1}{2}\right) + A_2 \sin^2\left(\frac{\omega_j T_2}{2}\right), j = 1, 2. \quad (10)$$

$$0 = T_1 A_1 \sin(\omega_j T_1) + T_2 A_2 \sin(\omega_j T_2), j = 1, 2. \quad (11)$$

FIG. 31 shows the symmetric, five-impulse sequence for two undamped modes. The impulse amplitudes do not depend upon the actual values of the frequencies, but only

upon r , the ratio of the frequencies. The impulse spacing is a function of the sum of the two frequencies. Furthermore, the impulses are evenly spaced as well as symmetric. This two-mode sequence should have several advantages over Singer's three-impulse sequences. First, the new sequence uses fewer impulses to cancel the same number of modes, while meeting the same constraints to increase insensitivity to modeling errors. Fewer impulses reduces the computation requirements for the impulse shaping, increasing the ease of implementation. Second, the two-mode sequence will always be shorter than convolved sequences. Comparing the length of the two sequences, we see that the length of the two-mode sequence is always less than the sequence resulting from the convolution of two *three-impulse sequences*.

$$\frac{4}{f_1 + f_2} \leq \frac{1}{f_1} + \frac{1}{f_2} \quad (12)$$

The time savings of the direct-solution sequence is up to 25% over the convolved sequence.

$$T_{\text{savings}} = \left(\frac{r-1}{r+1}\right)^2 \quad (13)$$

Unfortunately, the domain of frequency ratios, r , that yield valid sequences is limited. For frequency ratios greater than 3.0, this solution yields sequences with negative impulses. The five-impulse sequence has been generated for the two-mode, undamped system. Sequences can be generated for systems with significant damping as well, using the following equations, which were found by fitting curves to sequences generated by using Mathematica's Equation Solver [Wolfram, S., Mathematica™: A System for Doing Mathematics by Computer, New York: Addison-Wesley, 1988].

$$\begin{aligned} A_1 &= .1033 - .0899r + .0459r^2 + .2465\zeta + .2411r\zeta + .0155r^2\zeta \\ A_2 &= .0446 + .2903r - .1005r^2 + 1.5768\zeta - .9519r\zeta + .2216r^2\zeta \\ A_3 &= .7030 - .3964r + .1076r^2 - 1.4736\zeta + 1.5214r\zeta - .4604r^2\zeta \\ A_4 &= .0451 + .2846r - .0982r^2 + .3336\zeta - 1.4657r\zeta + .4953r^2\zeta \\ A_5 &= .1037 - .0884r + .0452r^2 - .6813\zeta + .6529r\zeta - .2714r^2\zeta \\ t_1 &= 0 \\ t_2 &= (.753 - .306r + .047r^2 + 2.017\zeta - 2.694r\zeta + .889r^2\zeta - 12.796\zeta^2 + 16.342r\zeta^2 - \\ &\quad 4.860r^2\zeta^2)t_d \\ t_3 &= (1.453 - .551r + .0788r^2 - .143\zeta + .232r\zeta - .0599r^2\zeta + 3.687\zeta^2 - \\ &\quad 4.543r\zeta^2 + 1.519r^2\zeta^2)t_d \\ t_4 &= (2.208 - .858r + .127r^2 + 2.052\zeta - 2.662r\zeta + .857r^2\zeta - 9.054\zeta^2 + 12.045r\zeta^2 - \\ &\quad 3.561r^2\zeta^2)t_d \\ t_5 &= (2.908 - 1.102r + .158r^2 + .757\zeta - .821r\zeta + .205r^2\zeta + 5.150\zeta^2 - 6.010r\zeta^2 + 2.001r^2\zeta^2)t_d \end{aligned} \quad (14)$$

where

$$r = \frac{\max(\omega_1, \omega_2)}{\min(\omega_1, \omega_2)}, \quad \zeta = \frac{\zeta_1 + \zeta_2}{2}, \quad t_d = \frac{2\pi}{\min(\omega_1, \omega_2)}$$

Using a similar derivation, a shorter sequence for two modes can be found. This sequence will be derived without using the derivative constraints (4) & (5). However, the resulting sequence will not be as insensitive to modeling errors as the five-impulse sequence. The three-impulse sequence for two modes has the same time spacing as the five-impulse sequence. The sequence amplitudes are as follows,

$$A_1 = A_3 = \frac{1}{4S} \quad (15)$$

$$A_2 = \frac{(2S - 1)}{2S} \quad (16)$$

where S is defined in FIG. 31.

When the damping ratios are not zero, finding solutions to the ten constraint equations is not as simple. Closed-form solutions have not been found. However, using GAMS, a non-linear equations solving software package, solutions can be found as described in [Hyde, J. and Seering, W., "Using Input Command Pre-Shaping to Suppress Multiple Mode Vibration," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA: April, 1991] and [Rappole, B. W.; Singer, N.C.; Seering, W.P. "Input Shaping™ with Negative Sequences for Reducing Vibrations in Flexible Structures," *Proceedings of 1993 Automatic Controls Conference*, San Francisco, CA].

Inensitivity to modeling errors

A major benefit of command shaping over other methods is its insensitivity to errors in modeling. The insensitivity of a sequence to errors in the estimate of system frequency can be calculated using the linear system response to a series of impulses, as given by eqn. (1). The insensitivity is measured as the range of system frequencies for which a sequence will reduce the residual vibration below 5% of the unshaped system response. Singer's three-impulse sequences have insensitivity of $\pm 13\%$ for each mode. A five-impulse sequence for 1.0 Hz and 2.25 Hz has insensitivity of $\pm 17\%$ around the lower mode and $\pm 8\%$ for the high mode as shown in FIG. 32. The high mode insensitivity will vary with the frequency ratio.

Experimental setup

The CMS is a four-axis robot used for handling work-in-progress during the manufacture of silicon wafers. FIG. 33 shows a drawing of the robot portion of the CMS.

The R-axis refers to the radial reach of the robot; the Z-axis is the vertical travel of the gripper and R-axis; the T-axis is the rotation of the whole robot. The W-axis is the rotation of the gripper. The R, Z, and T axes contribute significantly to the system vibrations. The W-axis motions do not tend to excite the system. The travel distance, direction, velocity and acceleration are shown in Table 7. The slow speed refers to the original controller. The fast speed refers to the speeds at which the Input Shaping™ tests were run. The robot stores cassettes of silicon wafers in racks surrounding the robot. There are two vibration problems with this system. First are structural vibrations found in the bending of the z-rail. These vibrations have significant amplitude and affect the positioning of the cassettes and the visual appeal of the system. Second are smaller vibrations in the r-axis that excite the wafers. Wafer rattle is of great concern as it generates particles that can contaminate the wafers.

Table 7: CMS Motion Parameters

Axis	Ran ge (in) *	Slo w Vel (in/ sec)	Slow Acce l (in/s ec ²)	Fast Vel (in/ sec)	Fast Acce l (in/s ec ²)
R (reach)	21	12.8	42.6	24.6	91.9
Z (up/do wn)	63	16.1	30.8	30.6	57.5
T (rotati on)	30 0	132	120	164	290

* Substitute degrees for inches on T.

A DSP (digital signal processing) based control system utilizing Input Shaping™ technology was developed for the robot. The controller is a PC-Bus based 4-axis board. A PID controller is run with a 1500 Hz servo rate. Encoder feedback is used in the control for both position and velocity. The velocity is calculated as the first difference in the encoder position. Desired position and velocity are commanded at each servo interrupt. The motions of the axes are based upon trapezoidal velocity profiles using the maximum velocity and acceleration values from Table 7. An unshaped move is a trapezoidal velocity profile. A shaped profile is the unshaped profile convolved with the shaping filter.

Before Input Shaping™ can be implemented, the system resonances must be determined. A laser interferometer was used to measure the vibrations in the CMS robot

during several standard maneuvers. These maneuvers were repeated with the laser retroreflector located at different points on the structure: on the z-rail and the r-axis endpoint. The system exhibits two primary resonances: 3.95 Hz and 5.10 Hz. These two frequencies are the bending of the z-rail in two directions. The frequency identification showed that these two frequencies remain within $\pm 5\%$ throughout the workspace. It was determined that a three-impulse sequence for these two frequencies be used. Preliminary tests with this sequence showed the presence of higher frequency vibrations that cause wafer rattle. Further identification revealed several smaller resonances at 6.7, 7.5, 8.7, 9.5, and 11.2 Hz. Two new sequences were calculated to remove these frequencies as well. The insensitivity plot for these two sequences are shown in FIG. 34. The 9-Impulse Sequence is a second three-impulse, two-mode sequence for 7.0 and 11.5 Hz, convolved with the 3-impulse sequence for the lower modes. The 15-impulse sequence is a five-impulse sequence for the higher modes convolved with the three-impulse sequence for the lower modes.

Two-Mode Shaper Experimental Results

Experiments were performed using the two sequences from FIG. 34. There were two goals in the experiments. The first goal was to validate the new sequences. The second goal was to determine if the speed of the robot could be increased while maintaining certain vibration tolerances.

The vibrations of the CMS are entirely unobservable at the feedback encoders. FIG. 35 shows a close-up of the settle portion of an unshaped, Fast, 20,000 count (16.78 in) R-axis maneuver. There are no vibrations present at the encoders. Similar results can be found on the Z and T axes.

In order to validate the new sequences, several R-axis maneuvers were performed. An unshaped Slow slew and unshaped Fast slew were compared to a 9-impulse shaped Fast slew, and a 15-impulse shaped Fast slew, as shown in FIGS. 36a, 36b, 36c and 36d, respectively. During these tests, the laser retroreflector was mounted on the side of the gripper. The laser only reads valid data at the end of the slew, when the retroreflector has entered the laser beam. All the laser data have been normalized such that the end position is zero. Table 8 contains the vibration reducing ability of the different sequences at different speeds. One common method of reducing vibrations is to slow down the machine. The data show that a 70% increase in move time results in at least 56% of the vibration from the original unshaped, fast move. Input Shaping™ produces superior results. By using the 15-impulse sequence, the move time is increased by only 44% and reduces the vibrations to less than 15% of the original magnitude.

Table 8: Comparison of R-Axis Moves to Fast Unshaped.

Seque nce Time (sec)	Spe ed Mov e Ti me (se c)	4 Hz (%)	5 Hz (%)	6.7 Hz (%)	7.5 Hz (%)
0.00	Fas t 5	10	10	10	10
0.33	Fas t 7	20. 9	20. 3	21. 5	18. 8
0.43	Fas t 7	18. 1	14. 3	10. 7	14. 6
0.00	Slo w 1	72. 1	58. 9	56. 0	66. 6

The new Input Shaping™ sequences are effective in reducing the vibrations of a particular move. In applying Input Shaping™ to the CMS, the goal was not to reduce vibrations, but to increase the throughput of the system. To this effect we must simultaneously decrease the move time and maintain the vibrations below the specified tolerance. Original, unshaped, slow moves are compared to Fast, shaped moves using both the 9-impulse and 15-impulse sequences. The shaped Fast speeds have been determined as acceptable in both positional accuracy and wafer rattle. Each move is tested on three axes: +16.78 in along the R-Axis, -26.25 in. along the Z-axis, and +211.26° clockwise in T-axis. Table 9 shows the results of these tests.

Table 9: Comparison of Moves: Fast 15 Impulse Shaped and Fast 9 Impulse Shaped to Slow Unshaped

a) R-Axis

Seque nce Time (sec)	Spe ed Mov e Ti me (se c)	4 Hz (%)	5 Hz (%)	6.7 Hz (%)	7.5 Hz (%)
0.33	Fas t 7	29. 0	34. 5	38. 5	28. 2
0.43	Fas t 7	25. 1	24. 3	19. 0	21. 9
0.00	Slo w 1	10 0.0	10 0.0	10 0.0	10 0.0

b) Z-Axis

Sequence Time (sec)	Speed	Move Time (sec)	4 Hz (%)	5 Hz (%)	8.7 Hz (%)	12.0 Hz (%)
0.33	Fast	2.16	70.9	29.4	12.43	42.9
0.43	Fast	2.25	51.0	33.8	8.8	42.3
0.00	Slow	2.70	10.0	10.0	10.0	10.0

c) T-Axis

Sequence Time (sec)	Speed	Move Time (sec)	4 Hz (%)	5 Hz (%)	9.0 Hz (%)	11.2 Hz (%)
0.33	Fast	1.71	73.3	31.6	78.5	83.0
0.43	Fast	1.81	80.2	31.5	12.3	82.7
0.00	Slow	2.15	10.0	10.0	10.0	10.0

In all cases, the vibration levels were reduced while the move time was also reduced by 15%- 21%. The T-axis shows the poorest vibration reduction, likely due to the poor resolution of the T-axis encoder. One encoder count of the T axis corresponds to 0.0029 in. in the direction of the laser. This poor resolution in T, limits the ability of reducing the vibrations to less than the the magnitude that the 15-impulse sequence does. The small decrease in the 3.9 Hz vibration during the Z-axis moves can be attributed to the small magnitude of the 3.9 Hz vibrations during unshaped Z-axis maneuvers. The lowest mode is barely excited during unshaped Z-axis moves; the 5.0 Hz mode is approximately 10 times greater in magnitude during Z-axis moves.

The tests also reveal that the 15-impulse sequence yields better performance at all frequencies than the 9-impulse sequence. The longer sequence is much more successful in reducing the wafer rattle. FIG. 37 shows an FFT of the laser data from an unshaped, Slow

R-axis slew and from a 15-impulse shaped, Fast slew. The Input Shaping™ yields an order of magnitude decrease in vibrations throughout the range important to the CMS.

Input Shapers™ for Digital Systems

Introduction

Real computer controlled systems have a limitation that is ignored in Singer and Seering's constraints for the three-impulse sequence, given by equations 2-8 in the previous section: the discrete time spacing of the controller. As depicted by the hash marks in FIG. 38, a computer controlled system operates using a time system which is discretized. However, both the three impulse sequence and the five-impulse sequence are essentially analog in nature and the exact locations of the impulses may not fall on the discrete time locations used by the computer controlled system. To fit the sequences to the digital spacing of a computer controlled system requires some alterations to the sequences. Several methods have been proposed. First, the impulses may be moved to the closest digital time step. This method will alter the frequency that the sequence will eliminate, but will maintain the insensitivity to modeling errors. Second, Watanabe and Murphy [Murphy, B. R., and Watanabe, I., "Digital Shaping Filters for Reducing Machine Vibration," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 2, April, 1992] suggest a graphical method for dividing impulses that fall between digital steps. The method they present will violate constraint (7), but normalizing the amplitudes to one can fix this problem. Third, Rappole presents a linear extrapolation method to divide impulses that fall between digital steps. This method works well when $f_{\text{sample}} / f_{\text{vibration}}$ is large. Fourth, Hyde and Seering [Hyde, J. and Seering, W., "Using Input Command Pre-Shaping to Suppress Multiple Mode Vibration," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA: April, 1991] and Rappole, Singer and Seering [Rappole, B. W. Jr., Singer, N. C., and Seering, W. P., "Input Shaping™ With Negative Sequences for Reducing Vibrations in Flexible Structures," *Transactions of the 1993 Automatic Controls Conference*, San Francisco, 1993] present a method of using GAMS (a non-linear equation solver) to find an exact solution to (2) - (8) with the digital time spacing also included in the constraints. The sequences from GAMS work very well, unfortunately, the optimization routines employed in GAMS are time consuming and not closed-form. FIG. 39 shows the insensitivity of sequences generated using the techniques listed above when used to digitize a sequence for 2.88 Hz. and 0.00 damping with a 10.0 Hz. sample rate. With these parameters, the linear extrapolation method works worst because of the low $f_{\text{sample}} / f_{\text{vibration}}$ value. The

sequence found by changing the time spacing of the three-impulse sequence has good insensitivity to modeling errors, but no longer eliminates the desired frequency.

Input Shaping Method

The last three methods are preferable to the first in that they generate sequences that eliminate exactly the desired frequency. All three of these methods are similar in that the sequence is changed from three impulses to five impulses, and all three use the two closest digital time steps to the impulses that fall between steps. We can express the times of a modified three-impulse sequence in equation form, where dt is the time step of the controller and t is the time spacing of the original sequence:

$$t_1 = 0.0 \tag{13}$$

$$t_2 = dt * \text{floor}(t/dt)$$

$$t_3 = t_2 + dt$$

$$t_4 = dt * \text{floor}(2*t/dt)$$

$$t_5 = t_4 + dt$$

By assuming that an exact solution to (2) - (8) exists with the five times given by (13), we now have five equations with five unknowns—the impulse amplitudes. We find the amplitudes from the matrix equation for the A_i :

$$A = M^{-1} V \tag{14}$$

Where

$$A = \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix}, \tag{15}$$

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ B_0 S_0 & B_1 S_1 & B_2 S_2 & B_3 S_3 & B_4 S_4 \\ B_0 C_0 & B_1 C_1 & B_2 C_2 & B_3 C_3 & B_4 C_4 \\ t_0 B_0 S_0 & t_1 B_1 S_1 & t_2 B_2 S_2 & t_3 B_3 S_3 & t_4 B_4 S_4 \\ t_0 B_0 C_0 & t_1 B_1 C_1 & t_2 B_2 C_2 & t_3 B_3 C_3 & t_4 B_4 C_4 \end{bmatrix}, \tag{16}$$

$$V = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (17)$$

$$B_i = e^{-\zeta \omega t}, \quad (18)$$

$$S_i = \sin(t_i \omega \sqrt{1 - \zeta^2}), \text{ and} \quad (19)$$

$$C_i = \cos(t_i \omega \sqrt{1 - \zeta^2}). \quad (20)$$

Using the above equations, we can calculate an exact solution for the Singer & Seering three-impulse sequence fit to the digital time spacing of a system. FIG. 40a shows the original three-impulse sequence from FIG. 39, while FIG. 40b shows the exact digital sequence from (14), which we will refer to as a five-impulse digital sequence. For many values of $f_{\text{vibration}}$, ζ , and f_{sample} , the new method yields the same sequence as GAMS will. The insensitivity of the five-impulse digital sequence will resemble that of the GAMS sequence in FIG. 39. Using a similar derivation, exact digital sequences for the five-impulse sequences derived above can also be found. When digitized, the five-impulse sequence will become a nine-impulse digital sequence. Using the equations presented previously for the damped five-impulse sequence, we calculate the digital times using the same idea as in (13). Once the nine times are calculated, the matrices in Equation (14) are expanded to 9×9 by repeating rows 2 - 5 for the second frequency, and adding extra columns for four extra impulses.

Digital Shaper Experimental results

Tests were conducted on a rotary table with two flexible beams as shown in FIG. 41. A PD controller was designed to control the position of the table, using the motor encoder for both position and synthetic tachometer feedback. The laser interferometer was used to measure the position of the thin, flexible beam, but it was not used for feedback. The laser and encoder were both sampled at the same rate (250 Hz.) by the computer controller. Under the PD control, the system showed a first frequency of vibration of 2.34 Hz. with approximately 0.10 damping ratio. The second vibratory mode has a frequency of 3.90 Hz., with approximately 0.01 damping ratio. In the configuration shown, the laser was able to measure the beam position through an angle of 30° . When measuring the residual vibration, the error due to misalignment caused by measuring the rotational vibration using a linear scale is less than 0.001 in. for vibrations with 0.250 in. amplitude. All maneuvers

use trapezoidal velocity profiles to avoid saturating the control signal. In the configuration shown, the rotary table has a velocity limit of 10 rad./sec. and an acceleration limit of 15 rad./sec./sec.

Tests were performed on the rotary table as shown in FIG. 41. Two different shaping sequences were compared with an unshaped maneuver. First, two five-impulse digital sequences were used, one for the 2.34 Hz. and 0.10 damping ratio mode and one for the 3.90 Hz. and 0.01 damping ratio mode. Second, one nine-impulse digital sequence for both modes was used. All sequences were calculated using the exact, digital solution method described above. FIG. 42 shows the position command and the encoder position for both the unshaped and the nine-impulse sequence shaped maneuvers from 1.0 rad. to 0.0 rad. Notice that only the 2.34 Hz. mode is visible at the encoder.

Theoretically, the two different shaping sequences should produce similar results. There are two differences that may make the nine-impulse digital sequence better than the two five-impulse digital sequences. First, the nine-impulse digital sequence is 0.644 sec. long, while the two five-impulse digital sequences take 0.692 sec., a difference of 7.5%. Second, when convolved together, the two five-impulse digital sequences result in one twenty-five impulse sequence; implementing twenty-five impulses is much more computationally intensive than is implementing one nine-impulse digital sequence.

FIGS. 43-45 show the laser measurement for the same maneuver from 1.0 rad. to 0.0 rad. The beam enters the range of the laser at 1.00 in. FIG. 43 shows the entire move; FIG. 44 shows the residual vibration of both the shaped and unshaped maneuvers; and FIG. 45 shows a close up of the residual vibration of the two shaped maneuvers. Table 10 shows a comparison of the vibration reduction achieved by the different shaping sequences. The percentages are calculated from the average of four maneuvers with each shaping sequence (or unshaped maneuvers). A 1024 point FFT is performed on the residual vibration and the magnitude of the FFT at the vibrational frequencies is measured.

Table 10: Comparison of Vibration Reduction for Nine-Impulse Digital Sequences and 2 @ Five-Impulse Digital Sequences.

Type of Sequence	Sequence Length (sec.)	2.34 Hz. Vibration (% of Unshaped)	3.90 Hz. Vibration (% of Unshaped)
Unshaped	0.000	100.00	100.00
Nine-Impulse Digital Sequence	0.644	0.40	7.71
2 @ Five-Impulse Digital Sequences	0.692	0.37	7.26

Input Shaping™ for Trajectory Following

Introduction

The effectiveness of Input Shaping™ for reducing residual vibration in point-to-point motions has been well established. However, very little work has been done to determine how Input Shaping™ effects trajectory following. Experiments in [Drapeau V.; Wang, D. "Verification of a Closed-loop Shaped-input Controller for a Five-bar-linkage Manipulator," *1993 IEEE Transactions on Robotics and Automation*] show a five-bar-linkage manipulator follows a clover pattern better with shaping than without. The shaping process alters the desired trajectory, so it is often assumed that Input Shaping™ will degrade trajectory following. While this may be true for temporal trajectories, trajectories where the location as a function of time is important, this section will show it is untrue for spatial trajectories, where only the shape of the move is important. Spatial trajectories compose a large percentage of the trajectory following applications including cutting, scanning, welding, machining and measuring.

One major reason the effects of Input Shaping™ on trajectory following have not been rigorously investigated is due to the difficulty in measuring the deviation from a desired trajectory. For this reason we studied a simple two-mode system and gave it uncomplicated, yet representative trajectories to follow.

Our model, shown in FIG. 46, represents a system with two orthogonal modes under PD control. The flexibility and damping of the controller and structure have been lumped together into a single spring and damper for each mode. The inputs to the system are x and y position commands. This model is very representative of gantry robots, coordinate measuring machines, and XY stages with flexible structures mounted to them. The following describes how Input Shaping™ affects the response to circular and square trajectory input, presents two simple methods for altering the unshaped command to better utilize Input Shaping™, and presents experimental results.

analytically a priori and the analytical solution can then be coded directly. The computer then need only evaluate the coded analytical solution rather than numerically calculating the full convolution, as is required in the general case. The advantage of this approach is a savings in computation time and memory storage. The memory storage requirements are reduced since it is no longer necessary to save the entire past command history and computation time is reduced because the many multiplications required in the convolution algorithm are no longer required. In the following, we will consider an example based on a trapezoidal trajectory. However, this technique can be extended to other trajectories which can be parameterized. The convolution may be performed using the parameterized function, yielding an analytical answer which is a function of the original parameters. The shaped input can then be calculated using the analytical answer, thus eliminating the need to store the previous command history. The original parameterization may include derivatives and integrals, and if, for example, the original input is simpler in acceleration, jerk or another derivative, then the shaping can be done on the simpler derivative and the result integrated by summation.

Continuing the trapezoidal trajectory example of FIG. 65, the shaped trajectory appears to be a complex curve. However, in acceleration, it is simply composed of two steps in acceleration and the effect of shaping the input is to produce an acceleration which has a staircase profile. Since the convolution operation is linear, shaping the original trajectory is equivalent to differentiating the trajectory to arrive at the acceleration profiles, shaping the acceleration profiles and then integrating the shaped acceleration profiles to arrive at the shaped trajectory profiles. The latter realization simplifies the computation in this case since the shaped acceleration profile is piece-wise flat. We need only store the transition times between the piece-wise flat elements and the values of each step and the shaped trajectory can then be calculated using only simple additions and branching statements. The following section presents the C code to implement a shaped trapezoidal velocity profile in this manner on an LM628/629 motion control chip.

Implementing Look-up Table for Shaped Trapezoidal Profiles on an LM628/629

The LM628/629 is set in position mode. To implement Input Shaping™, we recommend that the chip be serviced at a frequency, f_{sample} , which is greater than ten times the frequency being shaped, f . New position and velocity commands will be sent to the LM628/629 at f_{sample} .

Typically, the LM628/629 maximum acceleration is set to its maximum value. For a

system using a 6 MHz. clock on the LM628/629 and a 5,000 counts/in encoder, the maximum acceleration will be 10,000 in/sec². The maximum acceleration value is used, so that the LM628/629 is actually not doing any trapezoidal profiling. At each time step, we will command a maximum velocity and a desired position that will yield the maximum velocity for the entire command step.

The units used are encoder counts (count) for position, LM628/629 sample periods (sample) for time and command periods (step) for time. With a 6 MHz. clock, one sample is 341.33 μ sec. For calculations defining the trajectory, we use units of counts and steps. For commands being sent to the LM628/629, we use units of counts and samples.

Set Up: These operations are performed once upon system startup or they can be precalculated and hard coded.

- Required Parameters :

double accmax Maximum Acceleration of system. (in./sec.²)

double velmax Maximum Velocity of system. (in./sec.)

double f Frequency of vibration to eliminate. (Hz.)

double ζ Damping ratio of vibration.

- Define the conversion factors from inches to encoder counts and from command steps to LM628/629 sample periods.

double count_per_inch Convert from inches to encoder counts (count/in.)

double sample_period Sample Period, 6 or 8 MHz. Clock (sec./sample)

double sample_per_step Number of samples in a step. (sample/step)

```
#define count_per_inch 5000
```

```
#define sample_period 3.4133333 E-6 /* sec. for 6 MHz. Clock */
```

```
#define sample_per_step 50
```

```
#define step_period sample_per_step * sample_period /* 0.0170667 sec */
```

- Calculate the maximum acceleration and maximum velocity.

```
int32 vmax Maximum velocity (count/212 step)
```

```
int32 amax Maximum acceleration (count/216 step2)
```

```
vmax = (int)(velmax* count_per_inch * 212 * step_period);
```

```
amax = (int)(accmax* count_per_inch * 216 * step_period * step_period);
```

- Generate an Input Shaping™ sequence. A shaping sequence is a series of impulses, each specified by a time and an amplitude. The shaped input is calculated by convolving the shaping sequence with the raw input. The shaping sequence is calculated using the

method described in 'Generating Impulse Shaping™ Sequences for Digital Systems', a separate document. Typically, a shaping sequence is calculated only once for a particular system. The sequence is a function of the vibrational frequency f , damping ζ , and the sampling rate f_{sample} :

seq Digital shaping sequence.

int16 time[j] Time of jth pulse (step).
 int16 amp[j] Amplitude of jth pulse (dimensionless / 216).
 int16 pulses Number of pulses in seq.

For Each Maneuver. These operations are performed at the beginning of each maneuver:

• Required Parameters :

double xdes Desired final position. (in.)

• Recall the current command position :

int32 xo Current command position. (count/212)

• Calculate the move distance in encoder counts.

int32 xf Desired final position. (count/212)

int32 dx Change in position (count/212)

$xf = (\text{int})(xdes * \text{count_per_inch} + 0.5) \ll 12;$

$dx = xf - xo;$

• Calculate the unshaped trapezoidal profile values:

int16 Ssteps Sequence steps.

int16 Asteps Acceleration time steps.

int16 Bsteps Steps before deceleration.

int16 Tsteps Total Steps

int32 Aact Actual Acceleration (count/212 step²)

$Ssteps = \text{time}[\text{pulses} - 1];$

$Asteps = (\text{int})(\frac{v_{\text{max}} \ll 4}{a_{\text{max}}}) + 1 ;$

$Bsteps = (\text{int})(\frac{dx}{v_{\text{max}}}) + 1 ;$

if (Asteps > Bsteps) {

$Asteps = (\text{int})(\sqrt{\frac{dx}{a_{\text{max}}} \ll 4}) + 1 ;$

Bsteps = Asteps;

```
}
```

```
Tsteps = Asteps + Bsteps + Ssteps;
```

$$A_{act} = (\text{int}) \left(\frac{dx}{A_{steps} B_{steps}} \right);$$

- Generate a table of jerk values by multiplying the shaping sequence amplitudes, amp[i], by the actual acceleration, Aact, and changing units. An extra value is added to the table to prevent accessing values beyond the array bounds.

A trapezoidal profile can be described by four changes--ramp up to maximum positive acceleration, ramp down to zero acceleration, ramp down to maximum negative acceleration, and ramp up to zero acceleration. If the move distance is short, the profile will resemble a triangle, not a trapezoid, and the second and third changes in acceleration will occur at the same time. Each of these segments is a step in acceleration that is modified by the input shaping sequence. FIG. 66 shows how the superposition of four acceleration steps yields the desired trapezoidal trajectory. Each of the changes in acceleration is described by the jerk[] occurring at the shaping sequence time, time[], after the beginning of each segment of the profile:

```
int32    jerk[i] Acceleration Increment (count / 216 sample step).
```

```
for ( i=0 ; i<pulses ; i++ )
```

```
    jerk[i] = (amp[i]>>2) * (Aact>>2) / (sample_per_step<<8);
```

```
jerk[pulses] = 0;
```

```
time[pulses] = Tsteps;
```

- Initialize the variables used in implementing the shaped trapezoidal trajectory.

```
int32  acc      Current acceleration (count / 216 sample step)
```

```
int32  vel      Current velocity (count / 216 sample)
```

```
int32  pos      Current position (count / 212)
```

```
int16  i        Step Counter
```

```
int32  a        Command Maximum Acceleration (count / 216 sample2)
```

```
int16  stopflag  Flag for immediate stop.
```

```
/* The four indexes keep track of the four shaped acceleration steps */
```

```
int16  aindex = 0;
```

```
int16  bindex = 0;
```

```
int16  cindex = 0;
```

```
int16  dindex = 0;
```

```

acc = 0;
vel = 0;
pos = xo;
i = 0;
a = ((Aact<<4) / sample_per_step ) << 1; /* Double what we need */

```

Implement the shaped trapezoidal trajectory at the step frequency, *fstep*.

- At each time step, *i*, change the acceleration command, *acc*, increment *vel*, the desired velocity, and *pos*, the position command:

```

if( stopflag ) {
    if( i < Asteps )
        Asteps = i;
    if( i < Bsteps )
        Bsteps = i;
}
if( i == time[aindex] )
    acc += jerk[aindex++];
if( i == Asteps + time[bindex] )
    acc -= jerk[bindex++];
if( i == Bsteps + time[cindex] )
    acc -= jerk[cindex++];
if( i == Asteps + Bsteps + time[dindex] )
    acc += jerk[dindex++];
vel += acc;
pos += (sample_per_step * vel) << 4 ;
i++;

```

- Send the position and velocity commands, *x* and *v*, to the system.

```
int32    v    Velocity command to LM628/629 (count/216 sample)
```

```
int32    x    Position command to LM628/629 (count)
```

```
v = vel;
```

```
x = pos >> 12;
```

- After *Tsteps* steps, the shaped trapezoidal profile is completed:

Shaping Look-up Table for Arbitrary Commands

FIG. 67 shows the pullout torque–speed curve for a stepper motor. A trapezoidal trajectory is not the time-optimal trajectory for this motor because the torque speed curve is not constant. If a time-optimal trajectory is generated by looking up the maximum torque command at each speed, a given trajectory is defined. This trajectory best utilizes the abilities of the motor without attempting to exceed its capabilities. The resulting “optimal” trajectory can be shaped in advance for the system. A similar lookup table algorithm to the one shown above can be used to generate the optimal shaped profile.

Shaping During Emergency Situations

During emergency stopping situations shaping can also be useful. For some machines it is preferable to just command the current position and allow the system to vibrate. For some systems it is preferred to stop quickly without vibration. If shaping is used the system will reach a final position which is beyond the current position. The user must decide if it is more important to stop without vibration or to stop at the last position with overshoot. Because there is no overshoot with shaping, the final shaped position and the peak overshoot location are the same if a non-robust shaping sequence is used.

Shaping Stepping Motor Systems

Stepping motors and servo motors are commanded differently. To command a velocity with a stepper motor, a step or pulse rate is sent to the system [Leenhouts, Albert, Step Motor System Design Handbook, Litchfield Engineering Co., 1991] This pulse rate can be shaped just as a velocity command in a servo motor. Additionally, some stepper motor drive systems use vector control, thereby essentially commanding a torque from the motor. The proper coil is energized to deliver a torque. This energizing sequence can be shaped by using a shaping sequence to alter the shape of the desired torque profile and using the vector lookup table to translate the torque into a vector sequence.

Shaping controller product

Current commercial controller boards consist of a processor, encoder reading circuitry, and digital to analog conversion technology. Based on the input shaping technology described previously, a new hardware software product is possible. The product consists of a controller board with sensor reading circuitry, and frequency identification software. An external sensor (or a user supplied sensor) is connected to the system. A control signal is generated by the controller board. The system response from the sensor is read. The

resonances of the system are determined from the sensor data. These resonances are then used to set the shaping sequence.

Tuning Trapezoidal, S-Curve, and Parabolic Profiles for Vibration Reduction

Introduction

Several types of velocity profiles are used in the motion control industry to reduce vibration for point to point moves and improve tracking. The profiles are not based on the vibrational frequencies of the system, so the amount of vibration reduction obtained with the profiles varies from system to system. Additionally, the vibration reduction will vary with move distance. By taking the low mode of vibration and the move distance into account, we can tune Trapezoidal, S-Curve, and Parabolic velocity profiles for improved and predictable vibration reduction. Jones and Petterson [J.F. Jones and B.J. Petterson, "Oscillation Damped Movement of Suspended Objects," Proceedings of the IEEE International Conference on Robotics and Automation, 1988, Philadelphia, PA.] showed that a double acceleration pulse can damp a suspended object if the spacing is accurately set to one period of vibration. This result is essentially the Posicast solution. Petterson [B.J. Petterson and R.D. Robinett, "Model Based Damping of Coupled Horizontal and Vertical Oscillations in a Flexible Rod," AIAA Guidance, Navigation and Controls Conference, 1989, Boston, MA.] extended his earlier work by showing that the acceleration pulses of a trapezoidal trajectory can be accurately spaced or set to a specific duration to yield a vibration reducing trapezoidal acceleration.

Trapezoidal Velocity Profiles

A trapezoidal velocity profile is a ramp up at max acceleration (a_{\max}) to the maximum velocity (v_{\max}), slewing at v_{\max} for a specified amount of time, and then decelerating at maximum negative acceleration ($-a_{\max}$) to zero velocity. FIG. 68a shows a trapezoidal velocity profile and FIG. 68b shows its corresponding acceleration profile. Looking at the acceleration profile for the trapezoidal velocity profile, we see a pulse of acceleration of magnitude a_{\max} and length t_1 . Then zero acceleration, provided the move is long enough. At time t_2 , there is a pulse in acceleration of magnitude $-a_{\max}$ and length t_1 . The time required to reach maximum velocity (the time when v_{\max} is achieved) is:

$$t_1 = \frac{v_{\max}}{a_{\max}} \quad (1)$$

The time at which deceleration starts, t_2 , is given by:

$$t_2 = \frac{x_f}{v_{\max}} \quad (2)$$

where x_f is the desired travel distance. The total move time, t_3 , is given by:

$$t_3 = t_1 + t_2 \quad (3)$$

If we view trapezoidal profiles as step inputs in acceleration that have already been input shaped, we can deconvolve two two-impulse shapers out of the acceleration profile. That is, the acceleration profile of FIG. 69d is the convolution of the step function of FIG. 69a with the two two-impulse shapers shown in FIGS. 69b and 69c, and defined by

$$\text{Shaper 1} = \begin{bmatrix} \text{time} \\ \text{amplitude} \end{bmatrix} = \begin{bmatrix} 0 & t_1 \\ \sqrt{a_{\max}} & -\sqrt{a_{\max}} \end{bmatrix} \quad (5)$$

$$\text{Shaper 2} = \begin{bmatrix} \text{time} \\ \text{amplitude} \end{bmatrix} = \begin{bmatrix} 0 & t_2 \\ \sqrt{a_{\max}} & -\sqrt{a_{\max}} \end{bmatrix} \quad (6)$$

If we know the vibrational frequency, f , of the system in Hertz, we can calculate trapezoidal velocity profile metrics that are helpful for predicting residual vibration amplitude. The metrics are:

$$m_1 = t_1 * f \quad (7)$$

$$m_2 = t_2 * f \quad (8)$$

One, or both of these metrics must be an integer value for the trapezoidal profile to result in zero residual vibration.

We can use this information to develop guidelines that predict the effectiveness of trapezoidal velocity profiles. These guidelines are as follows:

Good Trapezoidal Velocity Profile Moves:

$$1) \quad m_1 = t_1 f = \frac{v_{\max}}{a_{\max}} f \equiv \text{integer value} \quad (9)$$

$$2) \quad m_2 = t_2 f = \frac{x_f}{v_{\max}} f \equiv \text{integer value} \quad (10)$$

Bad Trapezoidal Velocity Profile Moves:

$$1) \quad m_1 = t_1 f = \frac{v_{\max}}{a_{\max}} f \equiv \text{integer value} + 0.5 \quad \& \quad m_2 \neq \text{integer value} \quad (11)$$

$$2) \quad m_2 = t_2 f = \frac{x_f}{v_{\max}} f \equiv \text{integer value} + 0.5 \quad \& \quad m_1 \neq \text{integer value} \quad (12)$$

Physically, m_1 and m_2 are the number of cycles of vibration between the impulses in the two input shapers. If m_1 and m_2 are integers, the vibration at the system resonance will be

canceled. If m_1 and m_2 are integers + 0.5, the vibration at the system resonance is enhanced. As m_1 and m_2 become larger integers, more cycles of vibration occur before cancellation, and insensitivity to errors in the frequency estimate degrades. FIG. 70 shows this insensitivity loss by comparing the case of $a_{\max} = 2$ to the case for $a_{\max} = 1$, when $v_{\max} = 2$, $x_f = 3$ and $f = 1$. In terms of profile metrics, the first case has $m_1 = 1$ and $m_2 = 3$, while in the second case m_1 has been doubled to give $m_1 = 2$ and $m_2 = 3$. The loss of insensitivity to modeling errors also occurs when the move distance is lengthened (m_2 is increased). This is shown in FIG. 71, where m_1 is held constant at 1 while m_2 is increased from 3 to 6.

By examining the trapezoidal profile in this way, we gain a better understanding of its benefits and limitations. We can use the above metrics to evaluate moves in position, velocity, acceleration, or jerk, because zero vibration in any one of them necessarily means zero vibration in all the others. We can also use the metrics to set parameter values that will result in low vibration moves.

S-Curve Velocity Profiles

An S-Curve in velocity is a command of constant jerk (change in acceleration) up to the maximum acceleration, then constant acceleration is commanded, followed by a negative constant jerk to zero acceleration (trapezoidal in acceleration). This first component of the S-curve accelerates the system to a constant velocity, at which it slews for a specified amount of time. To decelerate, the system is given a negative constant jerk until the maximum negative acceleration is achieved, then constant maximum deceleration is commanded. Finally, a constant positive jerk is commanded to return the system to zero acceleration and zero velocity. The velocity, acceleration and jerk curves for an s-curve velocity profile are given in FIGS, 72a, 72b and 72c. The time required to reach maximum acceleration is given by:

$$t_0 = \frac{a_{\max}}{jerk} \quad (13)$$

The time at which constant acceleration ends is:

$$t_1 = \frac{V_{\max}}{a_{\max}} \quad (14)$$

The time at which deceleration starts, t_2 , is given by:

$$t_2 = \frac{x_f}{V_{\max}} \quad (15)$$

The total move time is given by:

$$t_3 = t_0 + t_1 + t_2 \quad (16)$$

If we view the S-Curve profile of FIG. 73e as the step input in jerk of FIG. 73a, that has already been input shaped, then we can deconvolve the three two-impulse shapers shown in FIGS. 73b, 73c and 73d.

The three two-impulse shapers are:

$$\text{Shaper 0} = \begin{bmatrix} \text{time} \\ \text{amplitude} \end{bmatrix} = \begin{bmatrix} 0 & t_0 \\ \sqrt[3]{\text{jerk}} & -\sqrt[3]{\text{jerk}} \end{bmatrix} \quad (17)$$

$$\text{Shaper 1} = \begin{bmatrix} \text{time} \\ \text{amplitude} \end{bmatrix} = \begin{bmatrix} 0 & t_1 \\ \sqrt[3]{\text{jerk}} & -\sqrt[3]{\text{jerk}} \end{bmatrix} \quad (18)$$

$$\text{Shaper 2} = \begin{bmatrix} \text{time} \\ \text{amplitude} \end{bmatrix} = \begin{bmatrix} 0 & t_2 \\ \sqrt[3]{\text{jerk}} & -\sqrt[3]{\text{jerk}} \end{bmatrix} \quad (19)$$

If we know the vibrational frequency, f , of the system in Hertz, we can calculate S-curve profile metrics that are helpful for predicting residual vibration amplitude:

$$m_0 = t_0 * f \quad (20)$$

$$m_1 = t_1 * f \quad (21)$$

$$m_2 = t_2 * f \quad (22)$$

As with trapezoidal profiles, if any of the above metrics are integers, the move will result in zero residual vibration. Additional insensitivity is achieved by making more than one metric an integer. Guidelines for predicting the effectiveness of S-Curves are:

Good S-Profile moves:

$$1) \quad m_0 = t_0 f = \frac{a_{\max}}{\text{jerk}} f \equiv \text{integer value} \quad (23)$$

$$2) \quad m_1 = t_1 f = \frac{v_{\max}}{a_{\max}} f \equiv \text{integer value} \quad (24)$$

$$3) \quad m_2 = t_2 f = \frac{x_f}{v_{\max}} f \equiv \text{integer value} \quad (25)$$

Bad S-Profile Moves:

$$1) \quad m_0 = t_0 f = \frac{a_{\max}}{\text{jerk}} f \equiv \text{integer value} + 0.5 \quad \& \quad m_1, m_2 \neq \text{integer value} \quad (26)$$

$$2) \quad m_1 = t_1 f = \frac{v_{\max}}{a_{\max}} f \equiv \text{integer value} + 0.5 \quad \& \quad m_0, m_2 \neq \text{integer value} \quad (27)$$

$$3) \quad m_2 = t_2 f = \frac{x_f}{v_{\max}} f \equiv \text{integer value} + 0.5 \quad \& \quad m_0, m_1 \neq \text{integer value} \quad (28)$$

To demonstrate the increase in insensitivity when more than one metric is an integer, FIG. 74 compares the case when $m_0=0.5$, $m_1=1$, and $m_2=3$ to the case when

$m_0=0.5$, $m_1=1$, and $m_2=3.5$. Case 1, where two metrics are integers, is significantly more insensitive than case 2, where only one metric is an integer.

Parabolic Profiled Trajectories

The velocity, acceleration and jerk profiles for a parabolic velocity profile are shown in FIGs. 75s, 75b and 75c. Parabolic velocity profiles are fundamentally different from Trapezoidal and S-Curve velocity profiles. Their frequency characteristics are from two four-impulse shapers that are added not convolved. This means that one metric from each shaper must be integers in order to have zero residual vibration.

To make the expressions for the parabolic velocity profile metrics compact, we introduce the following parameters:

$$p_1 = \frac{v_{\max}}{a_{\max}} \quad (29)$$

$$p_2 = \frac{a_{\max}}{\text{jerk}} \quad (30)$$

$$p_3 = \frac{x_f}{v_{\max}} \quad (31)$$

$$p_4 = \frac{(a_{\max})^3}{12v_{\max}\text{jerk}^2}$$

The important times at which changes in the profile occur are:

$$t_1 = p_1 - \frac{p_2}{2} \quad (32)$$

$$t_2 = p_1 + \frac{p_2}{2} \quad (33)$$

$$t_3 = p_3 - \frac{p_2}{2} \quad (34)$$

$$t_4 = p_3 + \frac{p_2}{2} \quad (35)$$

$$t_5 = p_3 + p_1 \quad (36)$$

Vibration reduction metrics can now be written as:

$$m_1 = t_4 f = \left(p_3 + \frac{p_2}{2} \right) f = \left(\frac{x_f}{v_{\max}} + \frac{a_{\max}}{2\text{jerk}} \right) f \quad (37)$$

$$m_2 = t_1 f = \left(p_1 - \frac{p_2}{2} \right) f = \left(\frac{v_{\max}}{a_{\max}} - \frac{a_{\max}}{2\text{jerk}} \right) f \quad (38)$$

$$n_1 = t_4 f = (t_3 - t_1) f = (p_3 - p_1) f - 0.5 = \left(\frac{x_f}{v_{\max}} - \frac{v_{\max}}{a_{\max}} \right) f - 0.5 \quad (39)$$

$$n_2 = p_2 f = \frac{a_{\max}}{\text{jerk}} f \quad (40)$$

Notice that 0.5 is subtracted from n_1 . This is because the impulses that occur at times t_1 and t_3 are both negative. With all other sets of two impulses, one was positive, the other negative. By subtracting 0.5 from this metric, we are insuring that the two negative impulses will cancel the vibration.

Guidelines for predicting the effectiveness of Parabolic velocity profiles are:

Good Parabolic Velocity Profiles:

$$1) \quad m_1 \equiv \text{integer value} \quad \& \quad \begin{cases} n_1 \equiv \text{integer value} \\ \text{or} \\ n_2 \equiv \text{integer value} \end{cases} \quad (41)$$

$$2) \quad m_2 \equiv \text{integer value} \quad \& \quad \begin{cases} n_1 \equiv \text{integer value} \\ \text{or} \\ n_2 \equiv \text{integer value} \end{cases} \quad (42)$$

Bad Parabolic Velocity Profiles:

$$\begin{array}{ll} 1) & m_1 \equiv \text{integer value} + 0.5 \quad \& \quad m_2 \equiv \text{integer value} + 0.5 \\ 2) & n_1 \equiv \text{integer value} + 0.5 \quad \& \quad n_2 \equiv \text{integer value} + 0.5 \end{array} \quad (43)$$

Procedures for Selecting the Parameters v_{\max} , a_{\max} , jerk for Trapezoidal Velocity Profiles

- 1) Set a_{\max} to $v_{\max} * f$
 - 2) Set a_{\max} to an integer times $v_{\max} f$
 - 3) Set v_{\max} to a_{\max} / f
 - 4) Set v_{\max} to an integer times a_{\max} / f
 - 5) Set v_{\max} to $x_f * f$
 - 6) Set v_{\max} to an integer times $x_f * f$
 - 7) Use any combination of the above 1), 2), 3), 4), 5), or 6) and convolve it with a shaper designed for the system frequency, or a shaper designed for the system frequency and damping ratio.
 - 8) If the move distance is such that the system's physical v_{\max} cannot be achieved, i.e. $v_{\text{actual}} = \sqrt{a_{\max} x_f} < v_{\max}$, and $\sqrt{x_f / a_{\max}} * f > 1$ then use the following procedure:
 - a) Define $n_v = \text{truncate}(\sqrt{x_f / a_{\max}} * f)$
 - b) Set $v_1 = a_{\max} n_v / f$ and $a_{\max} = a_{\max}$
 - c) Use $v_{\max} = v_1$, and $a_{\max} = a_{\max}$
- Or,
- d) Define $n_x = \text{truncate}(x_f / v_{\text{actual}} * f)$
 - e) Set $v_1 = x_f / n_x * f$ and $a_{\max} = a_{\max}$

f) Use $v_{\max} = v_1$ and $a_{\max} = a_{\max}$

The preferred command would be the shortest move time of the two moves. The move time for the trapezoidal profile is calculated as

$$\text{Trapezoidal move time} = v_{\max}/a_{\max} + x_f/v_{\max}$$

Procedures for Selecting the Parameters v_{\max} , a_{\max} , and jerk for S-Curve Velocity Profiles

All of the following procedures assume that we are starting with a system that has a known maximum velocity (v_{\max}), maximum acceleration (a_{\max}), move distance (x_f) and frequency of vibration (f). Given the above, we can modify the initial parameters to get vibration reducing moves.

Define: $n = \text{truncate}(v_{\max} * f / a_{\max})$

- 1) Set the jerk to be $a_{\max} * f$.
- 2) Set the jerk to be an integer times $a_{\max} * f$.
- 3) Set a_{\max} to $v_{\max} * f$
- 4) Set a_{\max} to an integer times $v_{\max} *$
- 5) Set v_{\max} to a_{\max} / f
- 6) Set v_{\max} to an integer times a_{\max} / f
- 7) Set v_{\max} to $x_f * f$
- 8) Set v_{\max} to an integer times $x_f * f$
- 9) If the move distance is nearly an integer times v_{\max} / f , use 1) or 2) and keep the acceleration at a_{\max} and the velocity at v_{\max} . This will result in a more robust move, and will be at the maximum acceleration of the system.
- 10) For additional insensitivity, use any combination of 1), 2), 3), 4), 5), 6), 7), 8), and 9)
- 11) Use 1) or 2) and if $v_{\max} * f / a_{\max}$ is less than $n(1 + v_{\max} / f / x_f)$ then set $v_{\max} = a_{\max} n / f$. If not, set $a_{\max} = v_{\max} f / (n + 1)$.

To further explain the above, we want the move that will cause no vibration and have the shortest move time. For S-Curve velocity profile moves, the move time is:

$$t_3 = t_0 + t_1 + t_2$$

Or,

$$t_3 = \frac{a_{\max}}{\text{jerk}} + \frac{v_{\max}}{a_{\max}} + \frac{x_f}{v_{\max}}$$

Using n defined above, we can calculate a new v_{\max} and a new a_{\max} :

$$v_{\max}^{\text{new}} = a_{\max} n / f$$

$$a_{\max \text{ new}} = v_{\max} * f / (n+1)$$

Calculate the new move times by replacing the velocity or the acceleration:

$$\text{movetime}_{v_{\max}} = a_{\max} / \text{jerk} + v_{\max \text{ new}} / a_{\max} + x_f / v_{\max \text{ new}}$$

$$\text{movetime}_{a_{\max}} = a_{\max \text{ new}} / \text{jerk} + v_{\max} / a_{\max \text{ new}} + x_f / v_{\max}$$

Select the shorter of the two movetimes and use the corresponding parameters. This will result in an s-curve velocity profile that will be more insensitive to modeling errors, while still maintaining a short move time.

Procedures for Selecting the Parameters v_{\max} , a_{\max} , and jerk for Parabolic Velocity Profiles

All of the following procedures assume that we are starting with a system that has a known maximum velocity (v_{\max}), maximum acceleration (a_{\max}), move distance (x_f) and frequency of vibration (f). Given the above, we can modify the initial parameters to get vibration reducing moves.

- 1) Set m_1 =intger value and set n_1 =integer value
- 2) Set m_1 =intger value and set n_2 =integer value
- 3) Set m_2 =intger value and set n_1 =integer value
- 4) Set m_2 =intger value and set n_2 =integer value

Input Shapers™ in Closed Loop Systems

For systems which are closed loops, an Input Shaper™ can be placed in the feed forward branch of the closed loop, as shown in FIG. 76, in order to modify or eliminate unwanted dynamics in the plant. Such a system is referred to as a shaper-in-the-loop (SIL) system and conventional control system design techniques can be applied to the SIL system if we consider the combination of the Input Shaper™ and the original (nominal) plant to constitute a new plant, as indicated by the dashed line in FIG. 76, and then design the controller with respect to the new plant. The new plant will have different magnitude and phase characteristics that may need to be considered when choosing SIL closed loop gains. Consider the following two cases.

Case 1: Shaped Frequency Greater than Closed Loop Bandwidth

If the shaped frequency is much greater than the nominal closed loop bandwidth, then there will be little effect on the nominal gain and phase margins. Little or no adjustment of the nominal gains will be required, and vibrations at the shaped frequency will be suppressed.

This technique is particularly valuable for suppressing vibration frequencies that are in a frequency band greater than the closed loop bandwidth and less than the Nyquist rate.

The shaper is tuned to the open loop vibration frequency to be suppressed. As the loop gains are increased, the locations of the shaper zeros and the system poles at the vibration frequency remain coincident.

The robustness property of the shaper improves under closed loop conditions. Poles tend toward zeros as loop gains are increased, consequently any misalignment of the shaper zeros with the poles at the vibration frequency is mitigated. Misalignment may be caused by errors in vibration frequency estimation, or by poles that change location with system configuration, with time, with payload, etc.

If the shaper zeros and poles at the vibration frequency are perfectly aligned, then little additional robustness should occur as a result of feedback signals circulating through the shaper. That is, disturbance rejection at the shaped frequency should not improve since the design of the shaper prevents feedback signals from generating the control output at the shaped frequency necessary to cancel the disturbance at the shaped frequency.

Case 2: Shaped Frequency within Closed Loop Bandwidth

If the shaped frequency is within or near the nominal closed loop bandwidth, then there may be a significant effect on the nominal phase and gain margins, thus requiring adjustment of the nominal gains (or modification of the filtering approach) in order to obtain a stable system with good transient response. Vibration at the shaped frequency will be suppressed, however, depending on the approach taken, the closed loop bandwidth may be degraded relative to the nominal system bandwidth depending on the amount of phase lag introduced by the shaper.

Therefore, for every closed loop system in which the input is shaped external to the closed loop, there should be an equivalent SIL system in which the shaping occurs inside the closed loop. Continuing along these lines, it is possible to make a SIL system mathematically equivalent to a system with a preshaped input. Consider the SIL system of FIG. 77, in which a shaper of transfer function S is inserted in the feed-forward branch of the loop and the inverse shaper S^{-1} is inserted in the feed-back branch. This system has the same overall system transfer function as the system of FIG. 78 in which the shaping occurs external to the closed loop. This is because the loop transfer function contains both S and S^{-1} , which cancel each other, resulting in the original loop dynamics for signals circulating in the loop; while reference inputs effectively pass through S only once. It is to

be understood that S is used here as a matter of convenience and is not meant to imply a specific type of transfer function. For example, in the case of digital systems, S may be the ratio of two z -transform polynomials; while in the analog case, S may represent a Laplace transform. Note that in an actual application using a digital controller, the cancellation would be implemented in a control computer. Since the implemented filter and its inverse are known exactly, the degree of cancellation depends only on the numeric precision of the computer.

Insertion of the inverse filter in the feedback branch eliminates the phase and gain modifications caused by the shaper inserted in the feed forward branch, effectively converting the SIL system to a prefiltered system. Therefore, for this implementation, the SIL filter is tuned to the closed loop vibration frequency to be cancelled.

Other filters besides S^{-1} may be inserted into the feedback path to partially (instead of totally) eliminate the phase and gain effects of the feed forward SIL filter. For example the filter $[(1-\alpha)+\alpha S^{-1}]$, where $0 < \alpha < 1$, can be inserted in the feedback branch. At $\alpha=1$, the phase and gain effects of the feed forward SIL filter are completely eliminated, and the system behaves as in FIG. 77, or equivalently FIG. 78. At $\alpha=0$, the system behaves as in FIG. 76 and the phase and gain effects of the feed forward SIL filter are unaltered. Thus, varying α from 0 to 1 effectively continuously varies the filter action from inside the loop to outside the loop.

Phase and gain effects due to a SIL filter inserted in the feed forward path can also be modified by modifying the SIL filter itself. Shorter length filters have decreased phase lag and gains that tend to increase with frequency. The shortest filters have negative going pulses that can approximate derivative action. Using a short length SIL filter may allow compensation of both the phase and gain curves resulting in increased bandwidth.

Miscellaneous Shaping Applications

The technology has been discussed primarily in the context of controlling the motion of a mechanical system, in general, and of a robotic arm with one or two modes of vibration, in particular. However, it should be understood that the technology is not restricted to these cases. It is also applicable to other types of dynamic systems, including electrical and chemical systems and to systems with more than two modes of vibration.

For example, to control the altitude of weather balloons, ballast is released to gain altitude and gas is vented to lower altitude. When altitude changes are made in this manner, the balloon will usually oscillate about the new altitude with a predictable frequency. The

altitude control effort can be shaped to greatly reduce oscillations in altitude. For example, a release of 10 lbs. of ballast shaped by a ZVD shaper would result in the dropping of 2.5 lbs. initially, then 5 lbs. after one half period of the vibration, and finally, the remaining 2.5 lbs. would be dropped after one period of vibration.

Finally, although the above examples have utilized impulse sequences, it should be understood that sequences of finite duration pulses may be used instead. The pulses or impulses can be constrained to begin and/or end at specified intervals, as is the case with impulses designed for use with systems which are discretized in time, and they may also be constrained to have specified amplitudes. In addition, pulses may be fixed in amplitude but variable in duration or fixed in duration but variable in amplitude.

FIG. 79 illustrates a standard implementation of the technology. An input is shaped using one of the methods described previously and the shaped input is used to drive the dynamic system. Shaping the input comprises convolving the input with a series of impulse sequences, or equivalently, as shown in FIG. 79, delaying and scaling the input by various time delays and gain factors and then summing the delayed and scaled inputs to form the shaped input. Different impulses sequences may be used, as disclosed herein, to achieve different characteristics for the shaped input or for the response of the dynamic system.

Figures

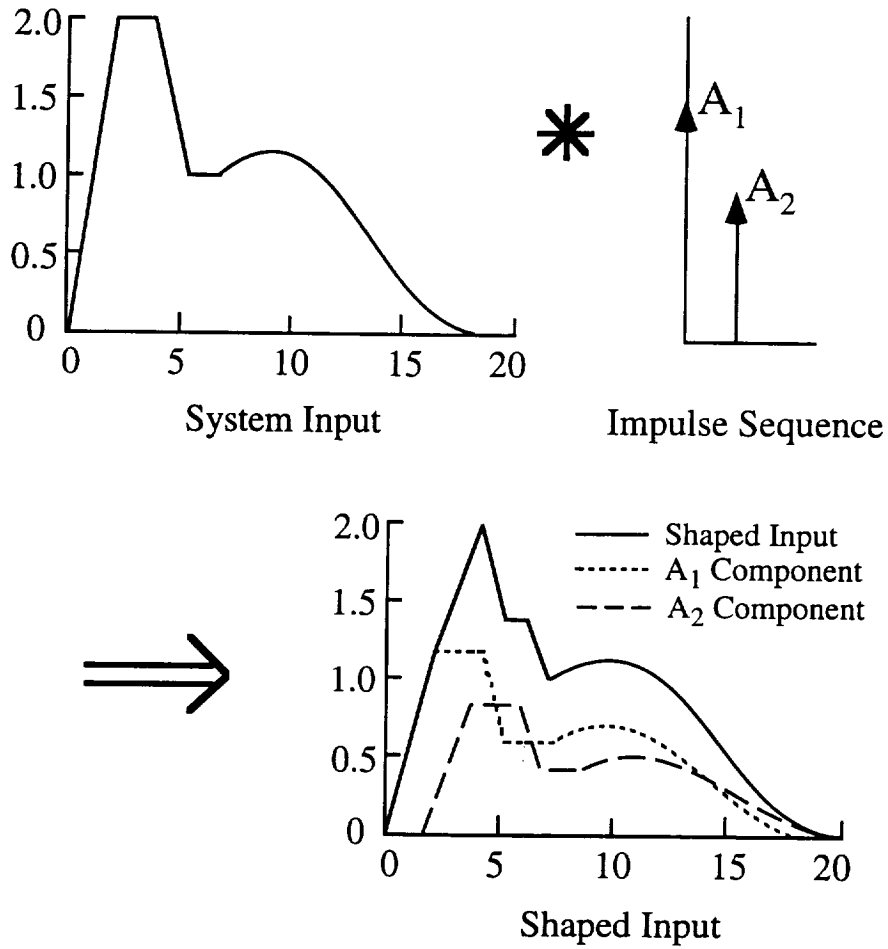


Figure 1 a, b, c: Convolution of a System Input with an Impulse Sequence Produces a Shaped Input

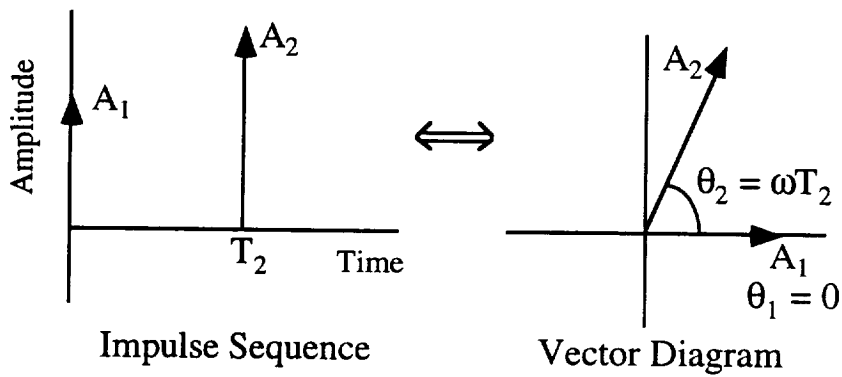
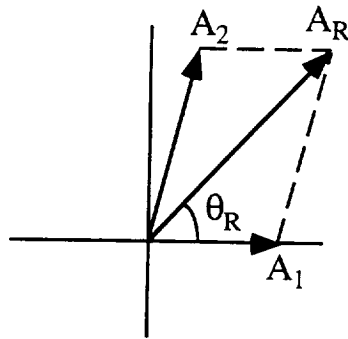


Figure 2 a, b: An Impulse Sequence and its Corresponding Vector Diagram



A_R = The amplitude of the vibration after impulses A_1 and A_2 have been applied to the system.

θ_R = The phase of the residual vibratic

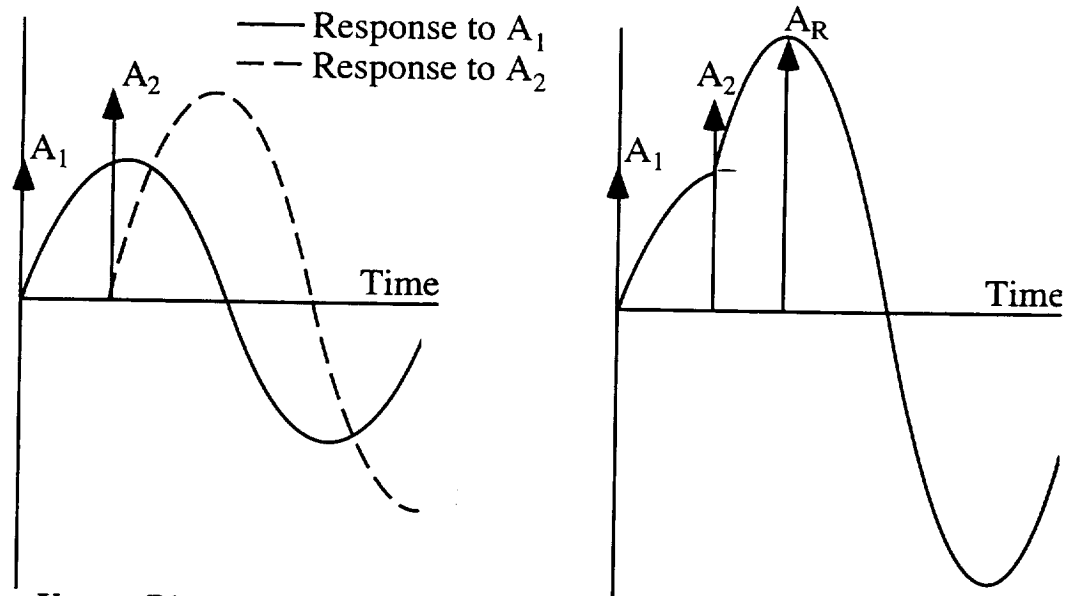


Figure 3 a, b, c: Vector Diagram and Time Domain Representations of Vibration.

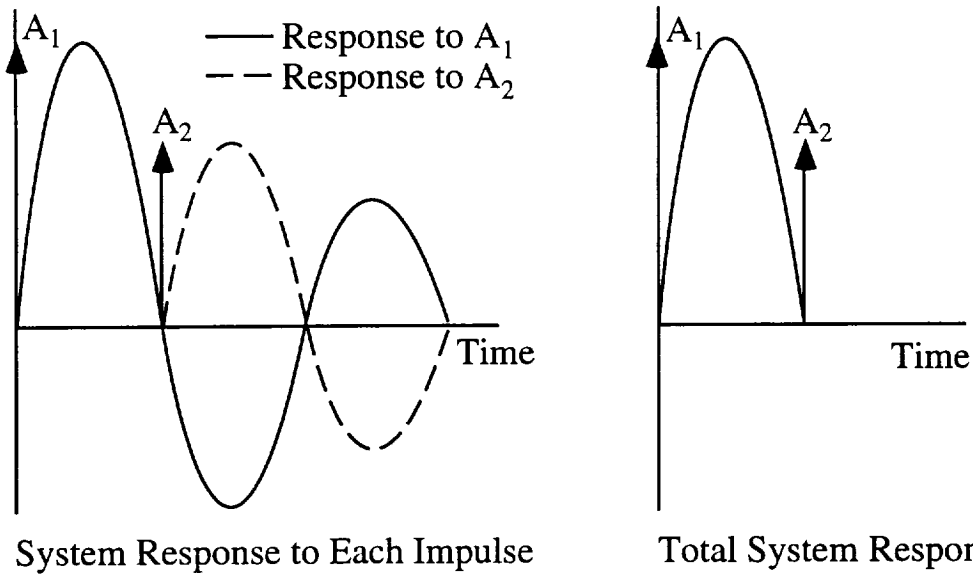


Figure 4 a, b: The Scaling Effect of Damping on the Canceling Vector.

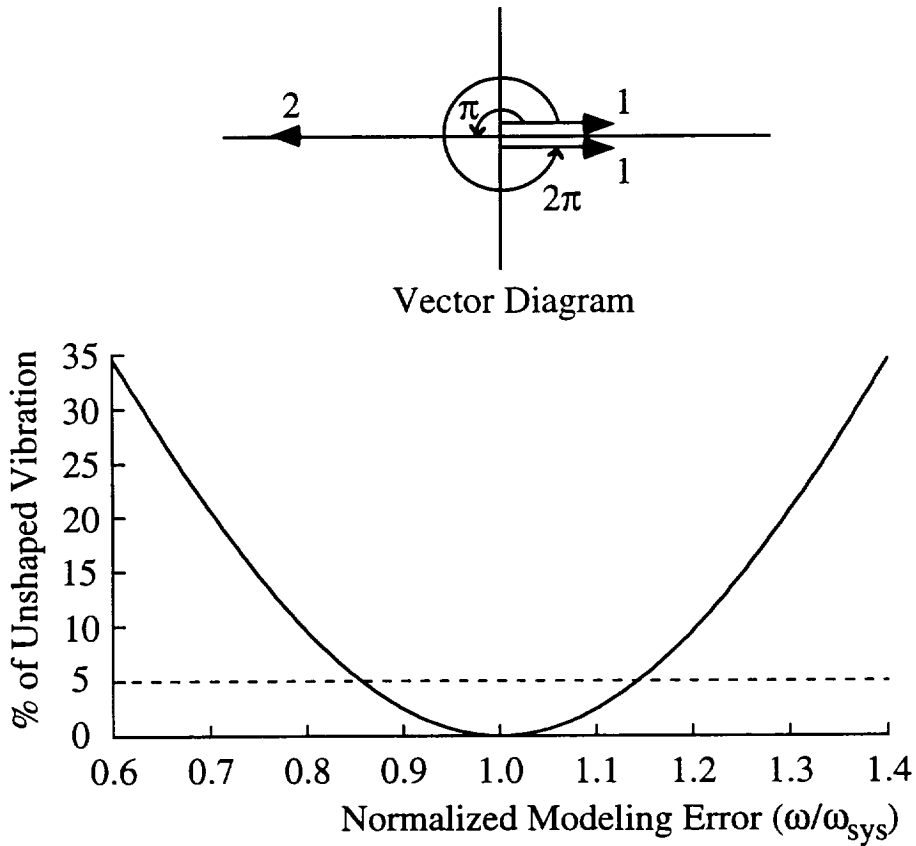


Figure 5 a, b: Vector Diagram of a Three-Impulse Sequence and its Sensitivity Curve.

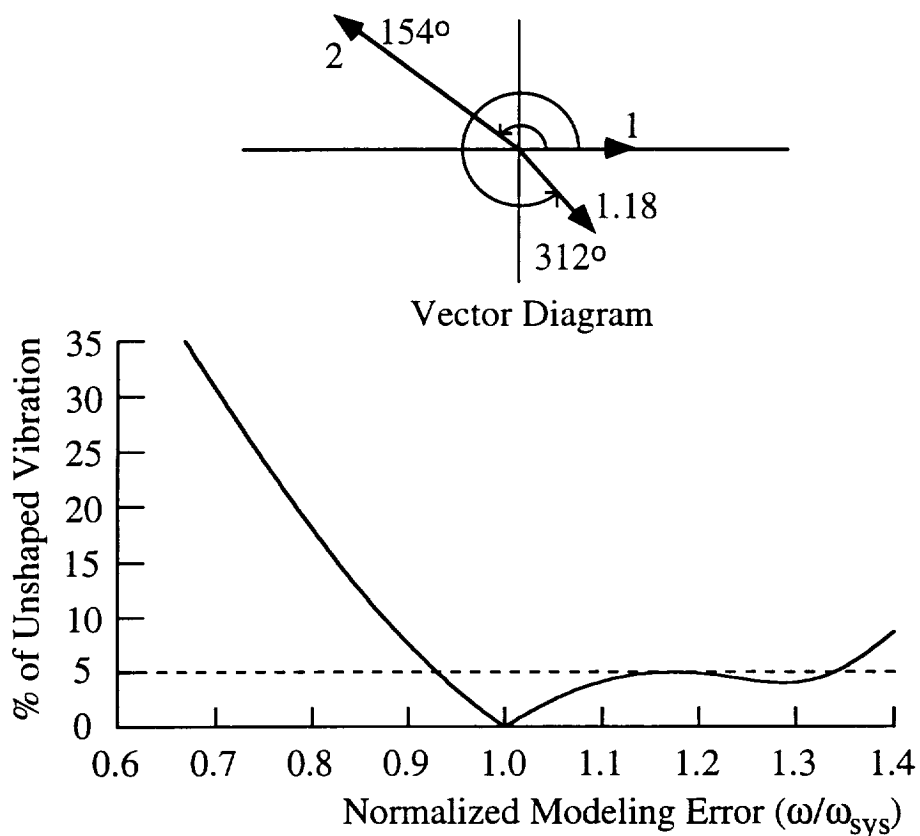


Figure 6 a, b: Vector Diagram and Sensitivity Curve of a Three-Impulse Sequence when the Second Vector is at 154 Degrees and the Third Vector Cancels the First Two.

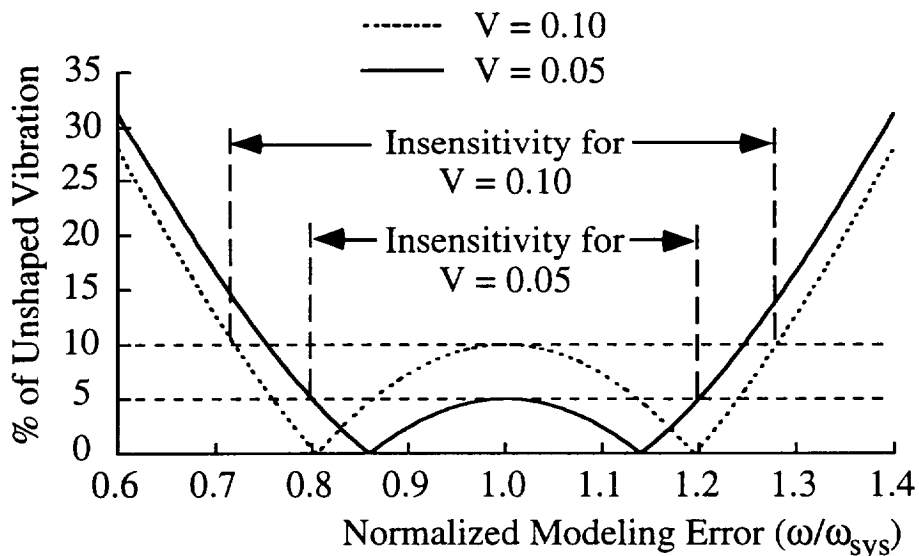


Figure 7: Sensitivity Curves for Vibration Limits of 5% and 10%.

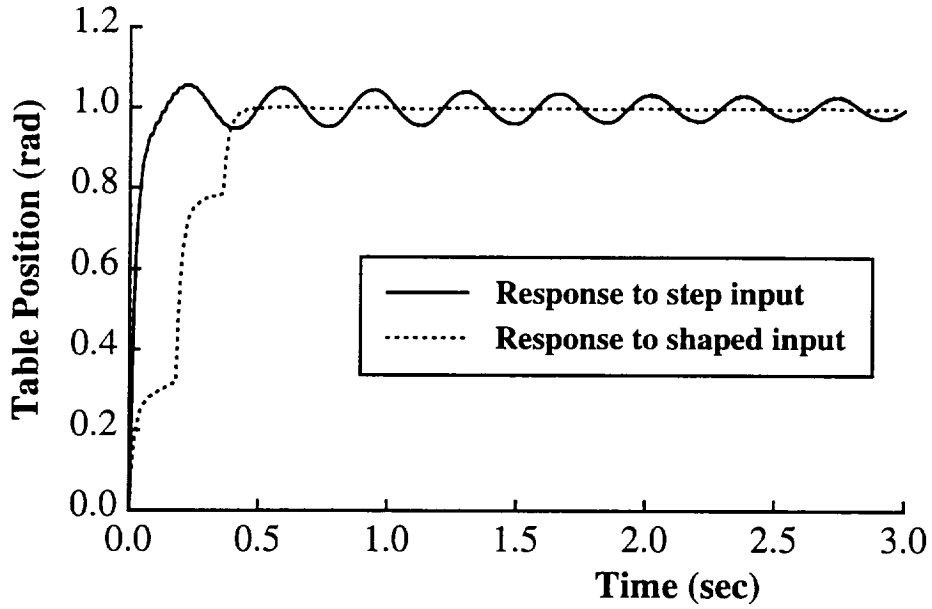


Figure 8: System Response to a Step Input and a Shaped Input

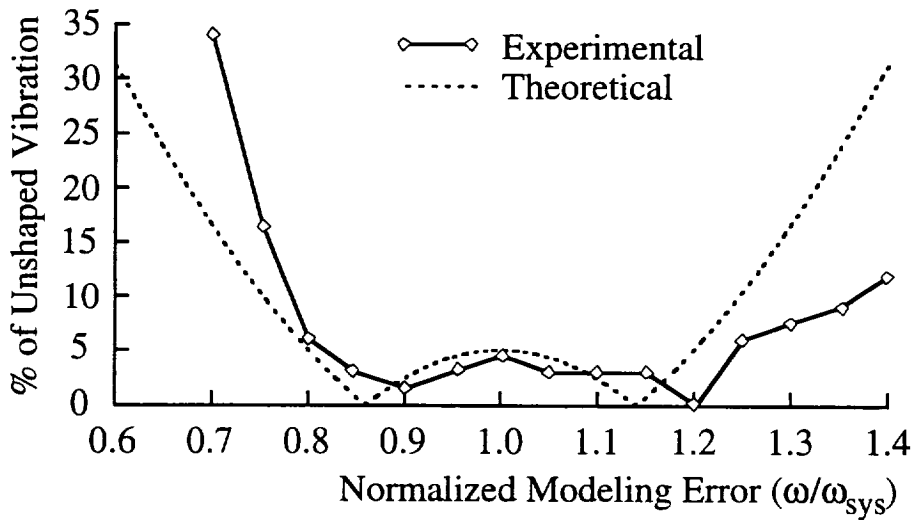


Figure 9: Experimentally Determined Sensitivity Curve for $V=0.05$.

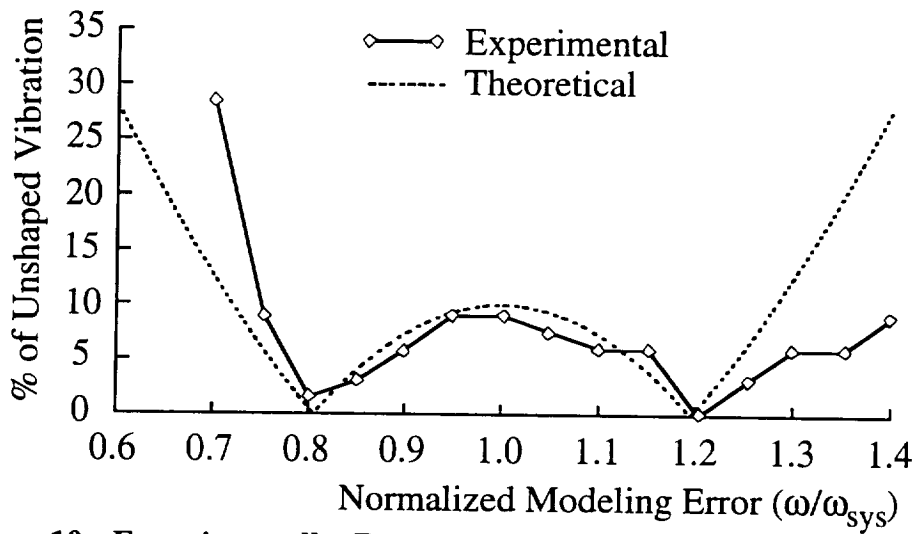


Figure 10: Experimentally Determined Sensitivity Curve for $V=0.10$.

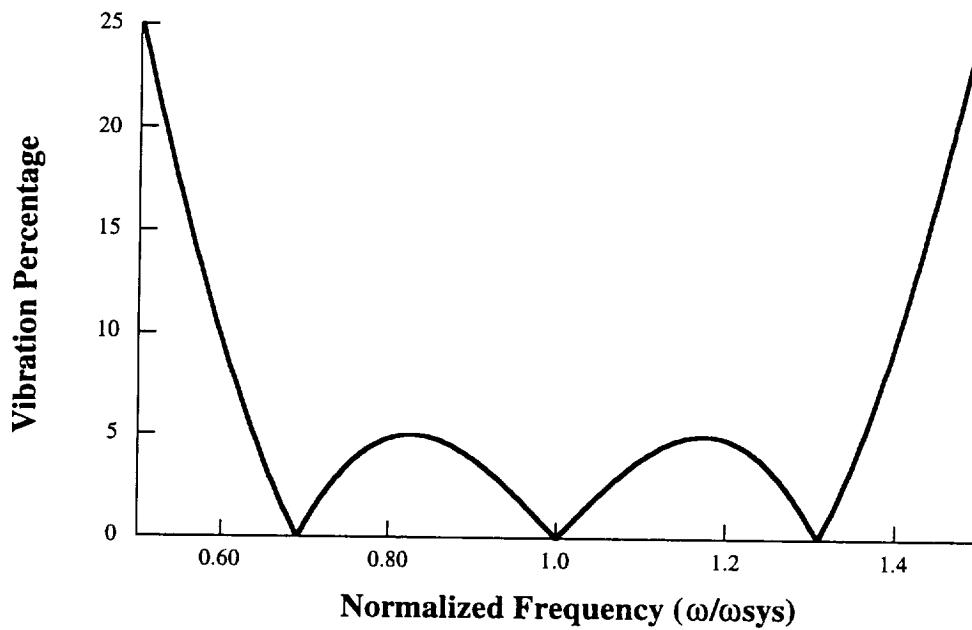


Figure 11: Sensitivity Curve for the Four-Impulse Extra-Insensitive Shaper

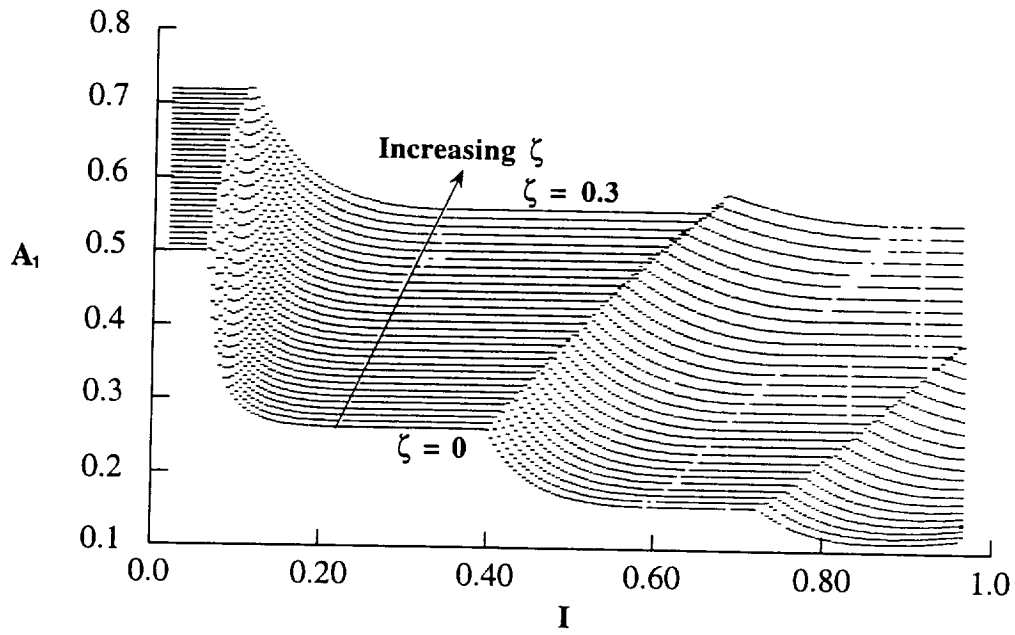


Figure 12: A_1 for the SI Shaper as a Function of I and ζ .

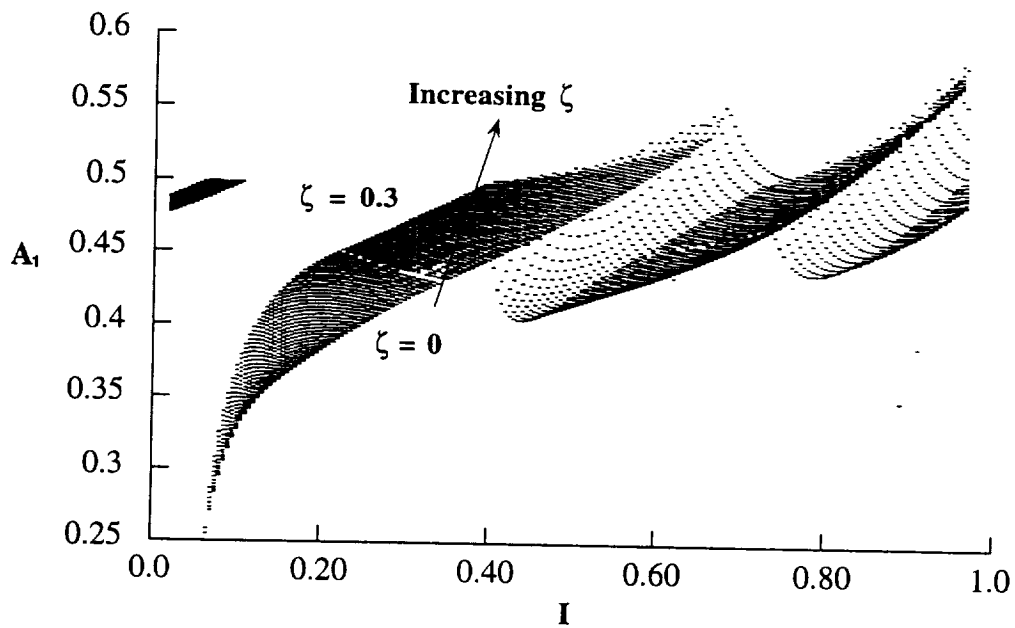


Figure 13: T_2 for SI Shaper as a Function of I and ζ .

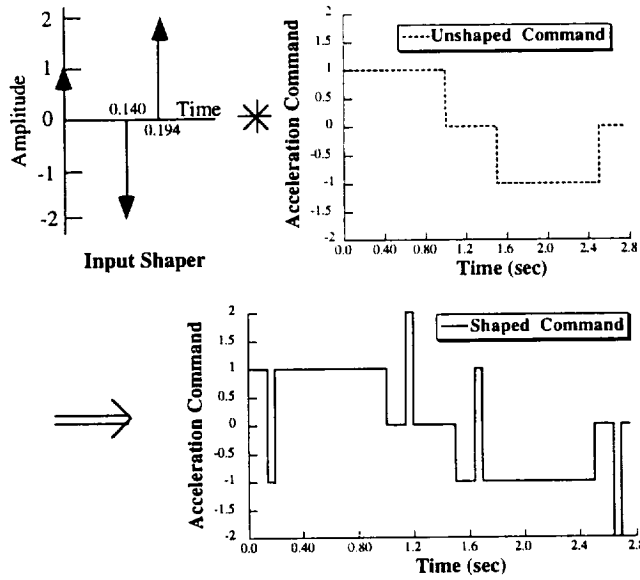


Figure 14 a, b, c: Over-Currenting Caused By Negative Input Shapers with P=1

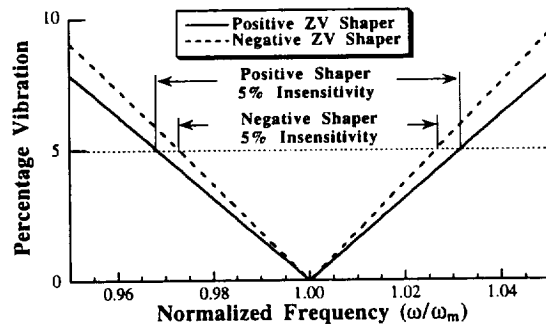


Figure 15: Sensitivity Curves for the Positive and Negative ZV Shaper with P=1

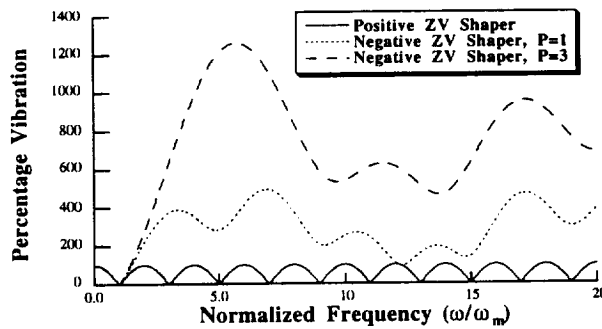


Figure 16: High-Mode Sensitivity Curves for the Positive and Negative (P=1&3) ZV Shapers

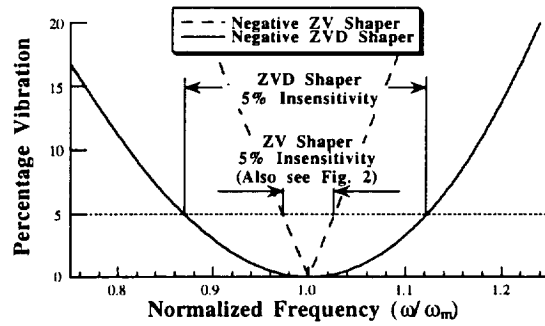


Figure 17: Sensitivity Curves for the Negative ZV and ZVD Shapers with $P=1$

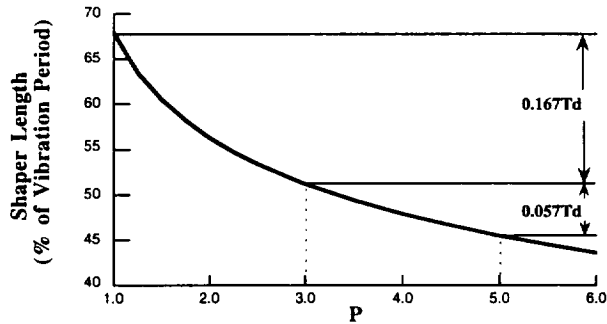


Figure 18: Length of Negative ZVD Shaper, $z=0$

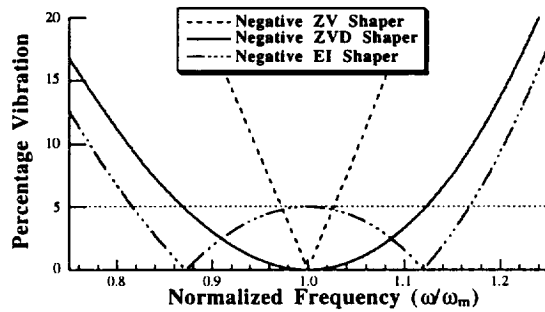


Figure 19: Sensitivity Curves for the Negative ZV, ZVD, and EI Shapers with $P=1$

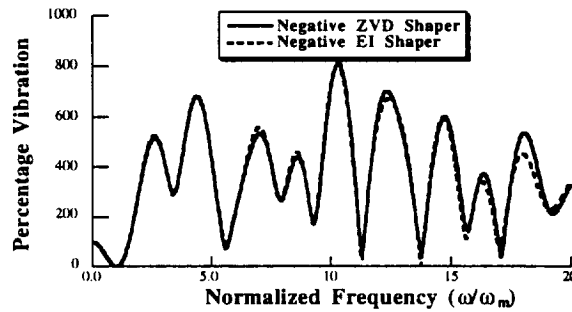


Figure 20: High-Mode Sensitivity Curves for the Negative ZVD and the $V=5\%$ EI Shapers

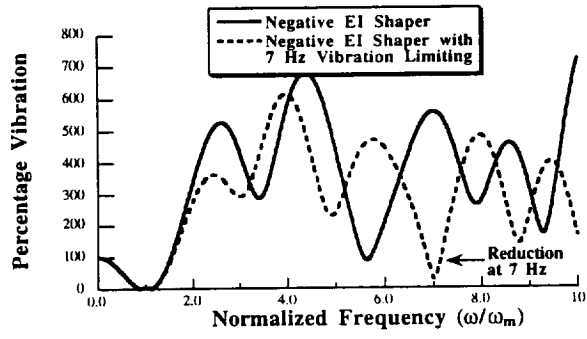


Figure 21: Effect of 7 Hz Vibration Limiting

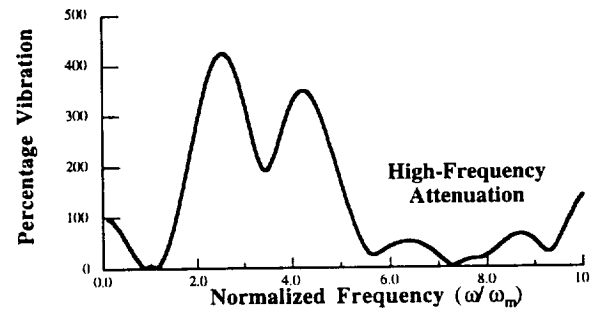


Figure 22: Sensitivity Curve for the EI Shaper with Low-Pass Filter

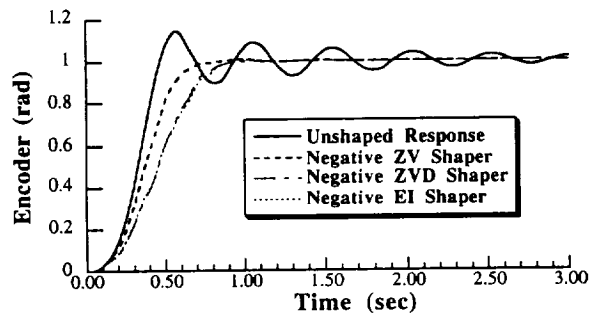


Figure 23: Plots of Unshaped and Shaped Responses

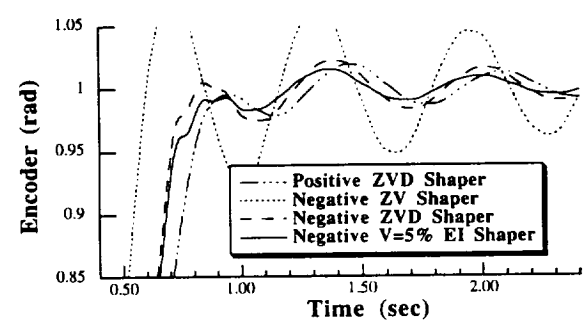


Figure 24: Zoom in on Residual Vibration When Frequency is 25% Lower than Modeled

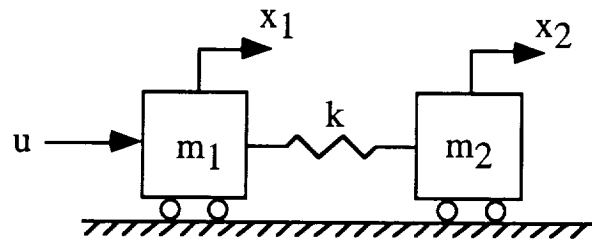


Figure 25: Simple System Model

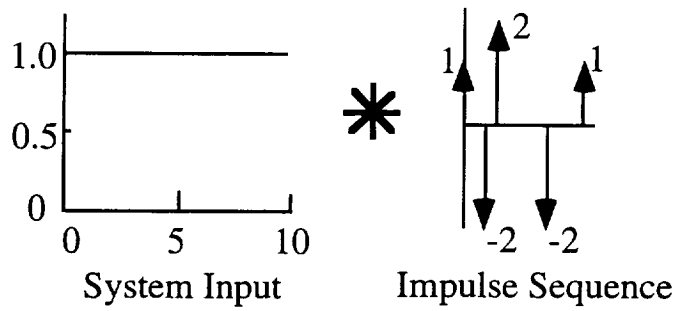


Figure 26 a, b, c: The Shaping Process for Creating Constant-Magnitude Pulses

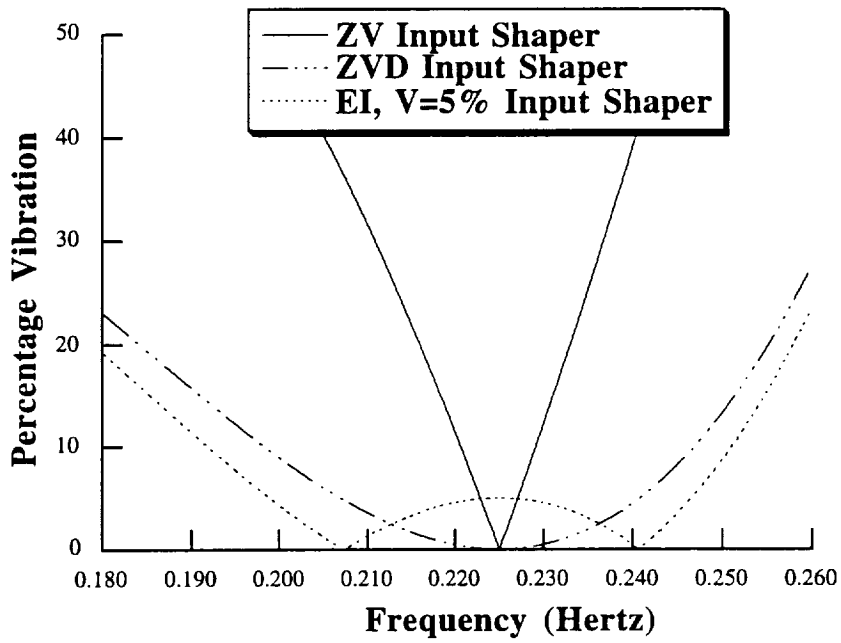


Figure 27: Comparison of the Sensitivity Curves for the ZV, ZVD, and EI Constant-Magnitude Shapers

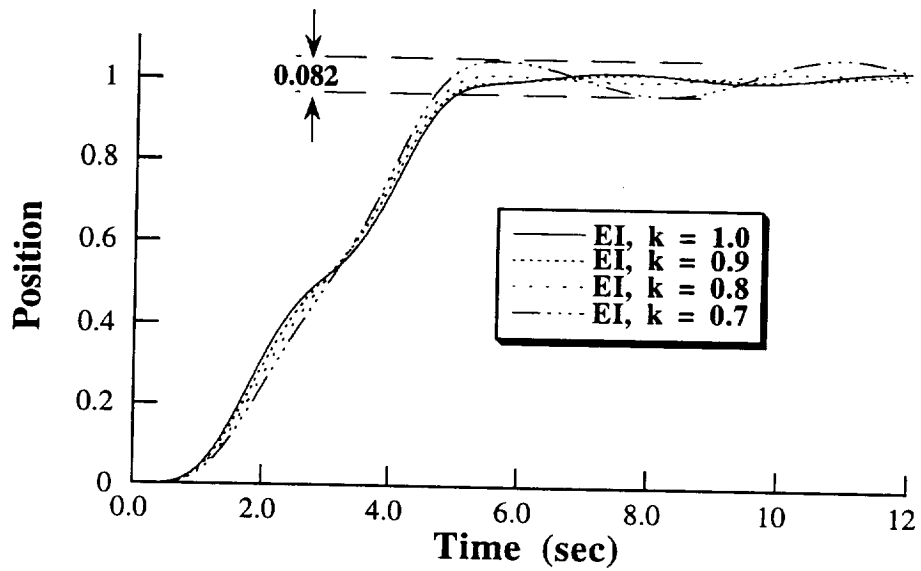
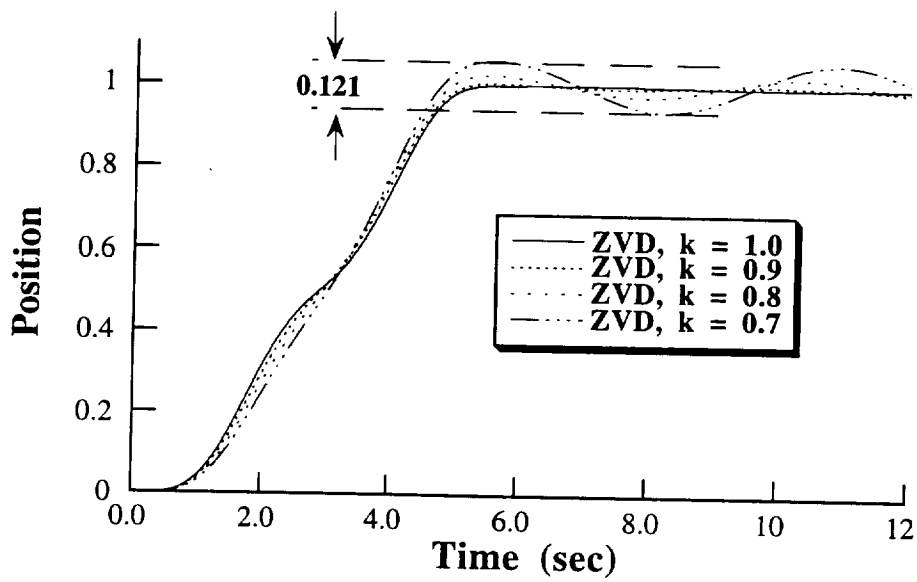


Figure 28 a, b: Comparison of Responses to ZVD and EI Shapers for Various Values of k

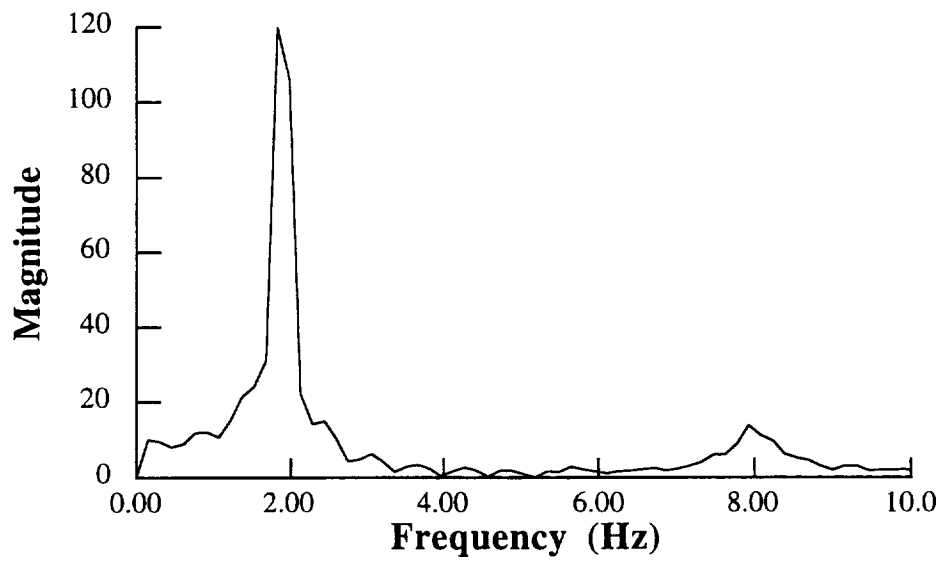


Figure 29: Frequency Spectrum of Unshaped Vibration

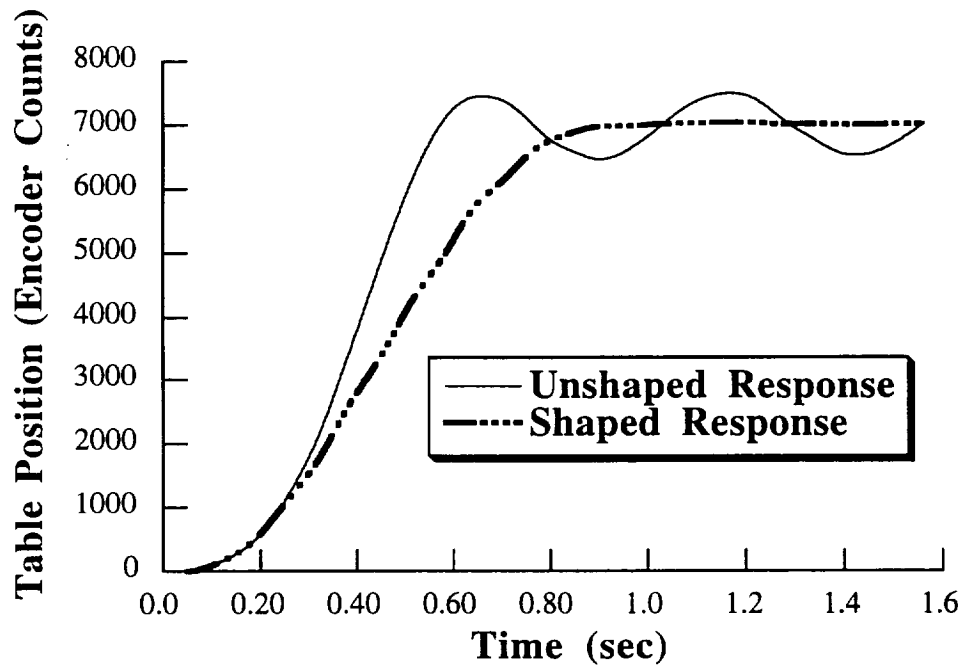


Figure 30: Comparison of Unshaped and Shaped Responses

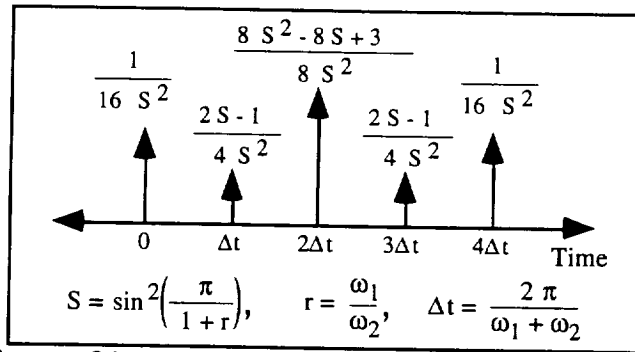


Figure 31: Two-Mode, Symmetric Sequence.

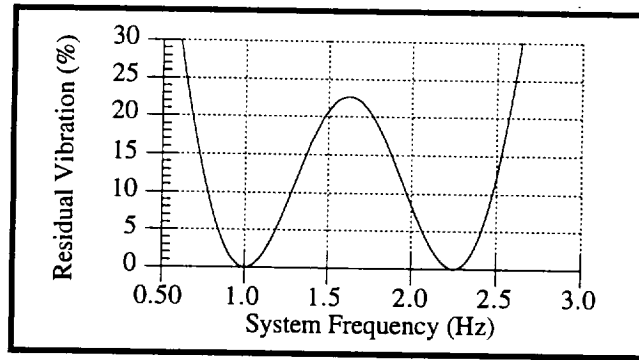


Figure 32: Insensitivity of Five-Impulse Sequence for 1.0 and 2.25 Hz.

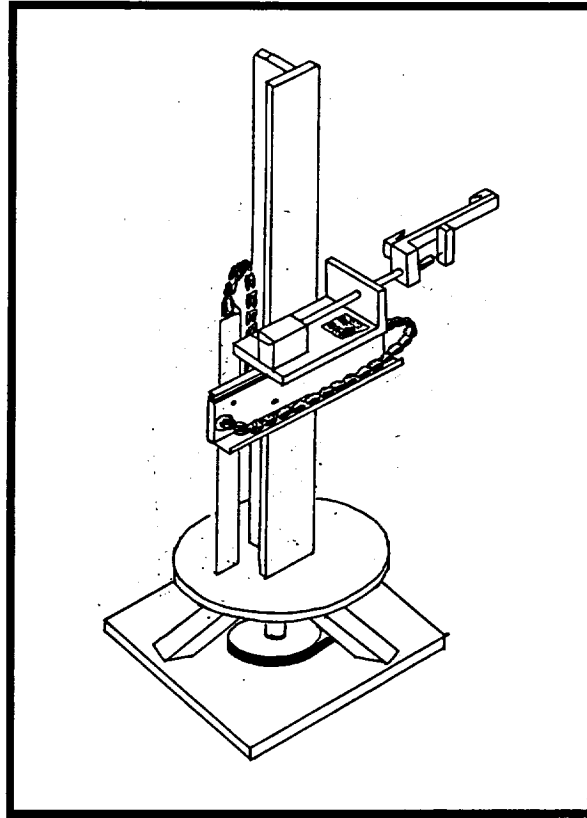


Figure 33: The Proconics CMS.

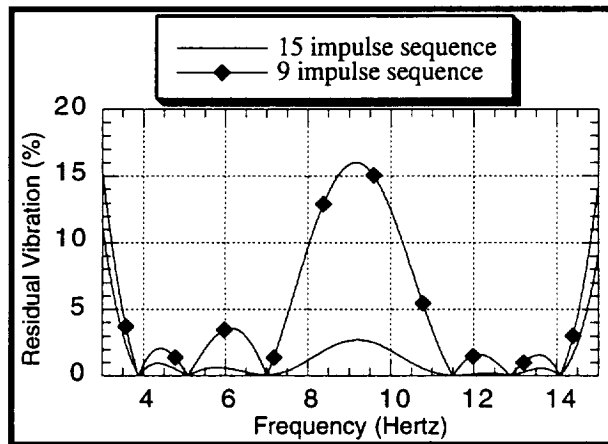


Figure 34: Insensntivity of 9 and 15 Impulse Sequences.

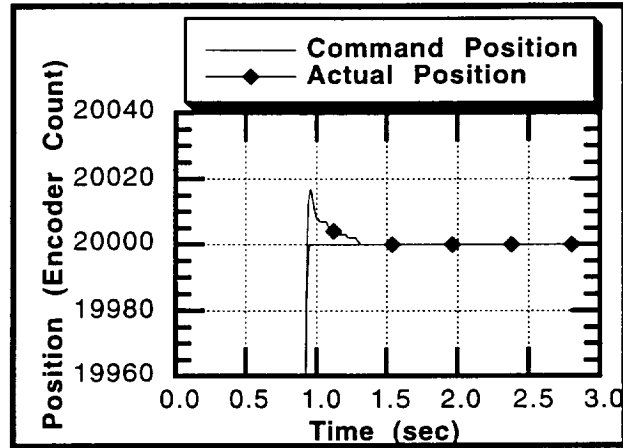
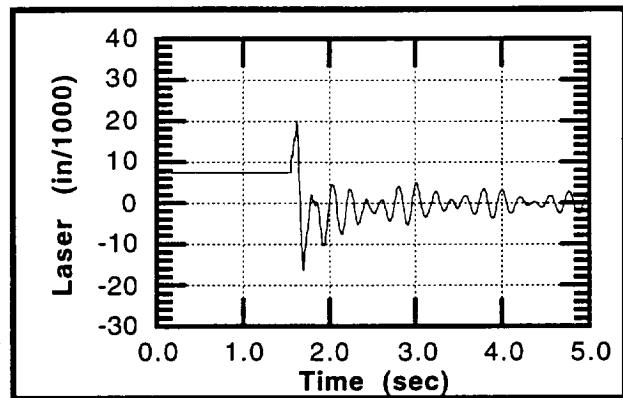
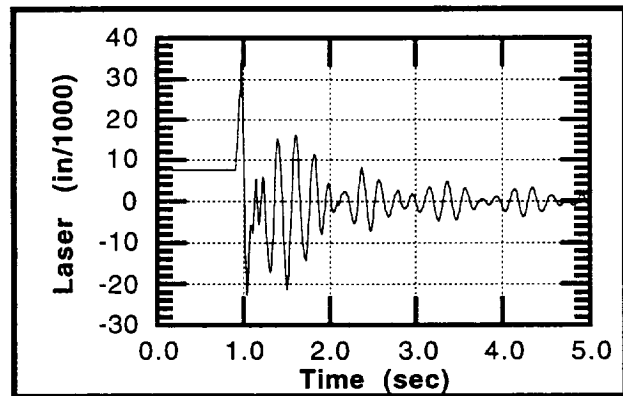


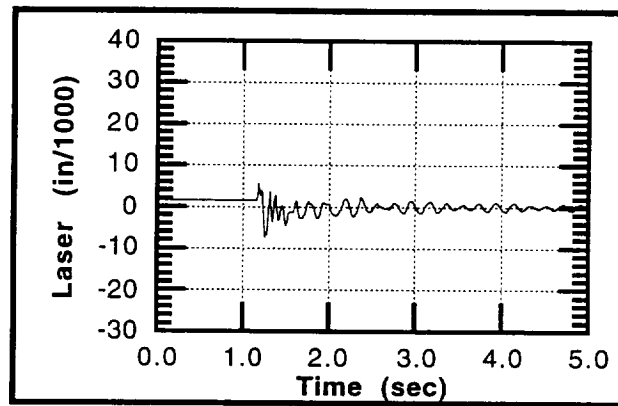
Figure 35: Encoder Position for R-Axis Slew. (detail)



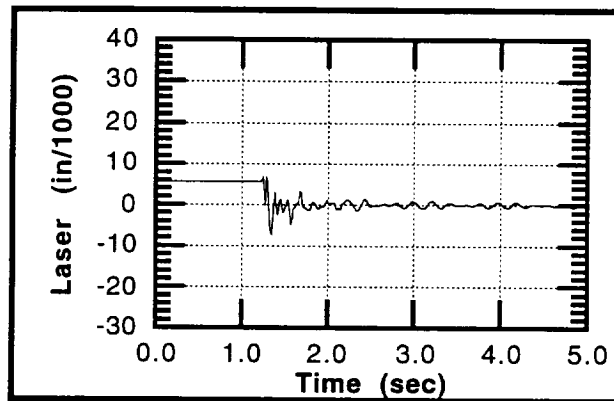
a) Original Response—No Shaping, Slow Speed.



b) Unshaped Response—Fast Speed.



c) Shaped Response–Fast Speed, 9-Impulse Sequence.



c) Shaped Response–Fast Speed, 15-Impulse Sequence.

Figure 36: Unshaped and Shaped Response.
 +16.78 in. R-Axis Moves.
 Retro Mounted on Side of Gripper.

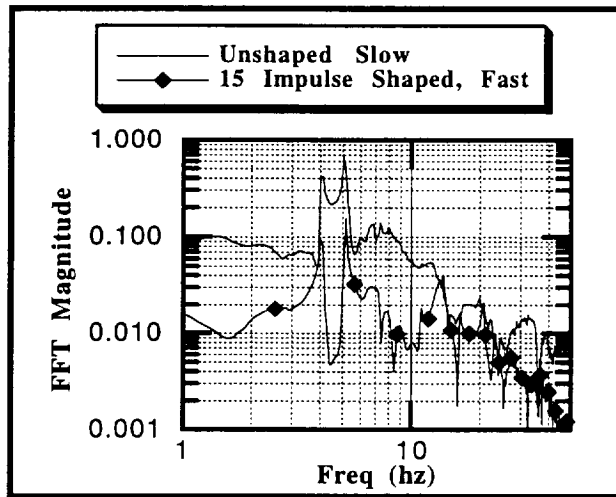


Figure 37: FFT Magnitude Comparison.
 +16.78 in. R-Axis Move
 Retro Mounted on Side of Gripper.

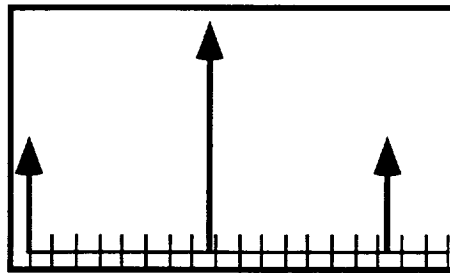


Figure 38: A Three-Impulse Sequence May Not Fit the Digital Time Spacing of the System

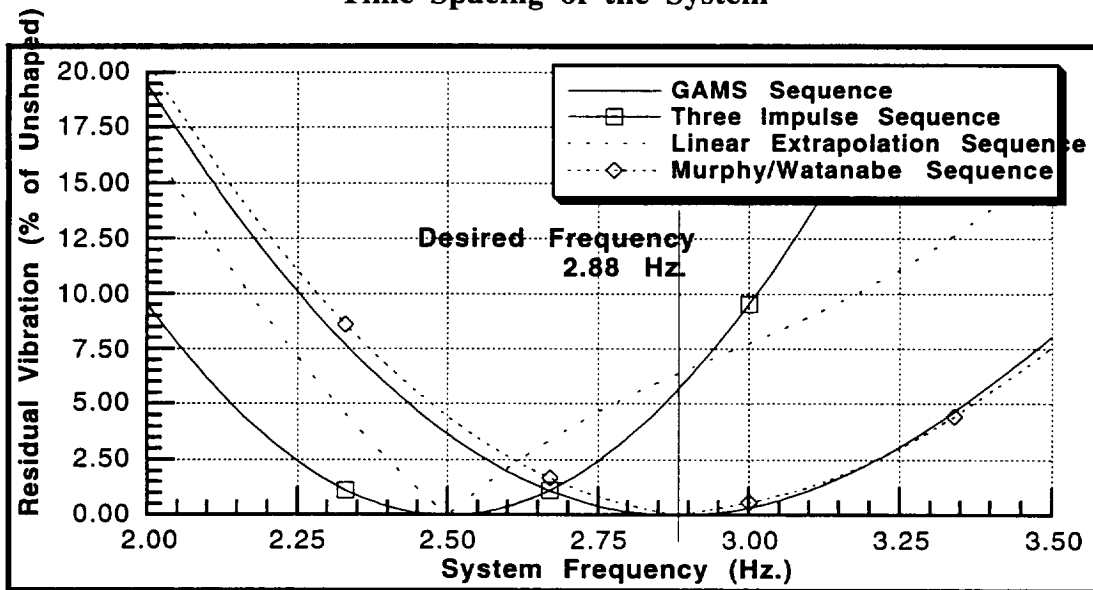
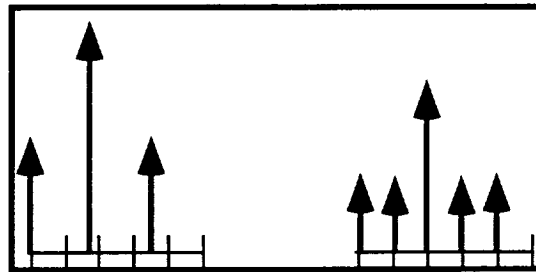


Figure 39: Comparison of Sequence Digitizing Methods.
 $f_{\text{vibration}} = 2.88 \text{ Hz.}$, $\zeta = 0.00$, $f_{\text{sample}} = 10.0 \text{ Hz.}$



Three Impulse Sequence Matrix Inversion Sequence

Figure 40 a, b: Three Impulse Sequence and Sequence Found By Matrix Inversion for $f_{\text{vibration}} = 2.88 \text{ Hz.}$, $\zeta = 0.00$, $f_{\text{sample}} = 10.0 \text{ Hz.}$

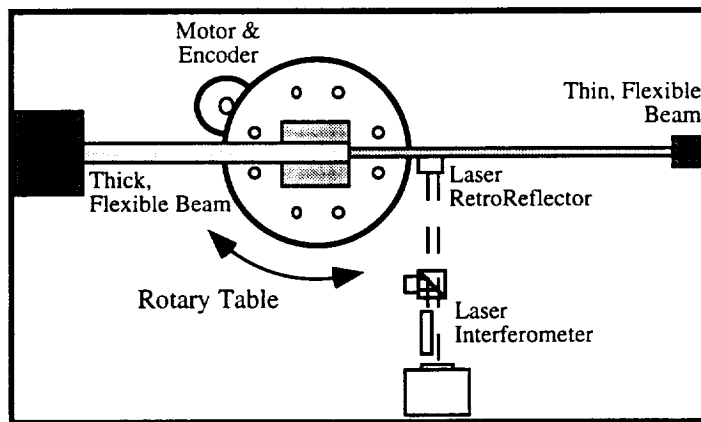


Figure 41: Experimental Set Up for Tests on the Rotary Table With Two Modes of Vibration.

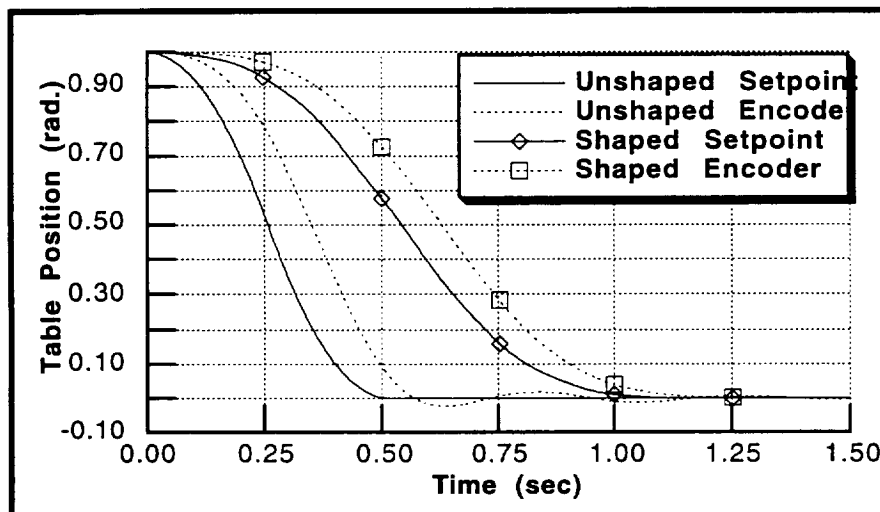


Figure 42: Position Set Point and Encoder Position for Shaped and Unshaped Maneuvers.

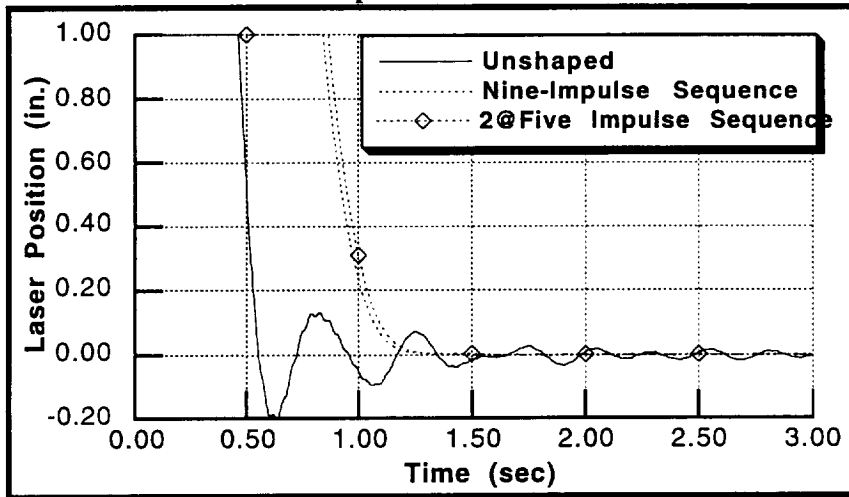


Figure 43: Laser Position of Unshaped and Shaped Maneuvers.

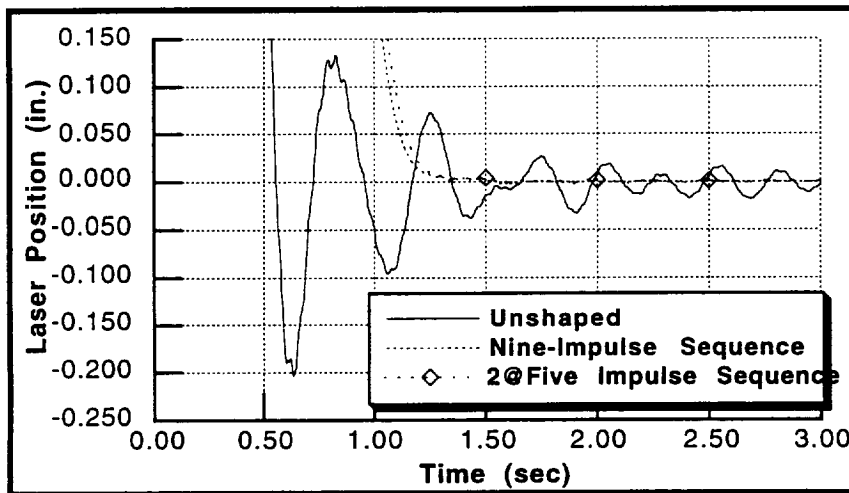


Figure 44: Residual Vibration of Unshaped and Shaped Maneuvers.

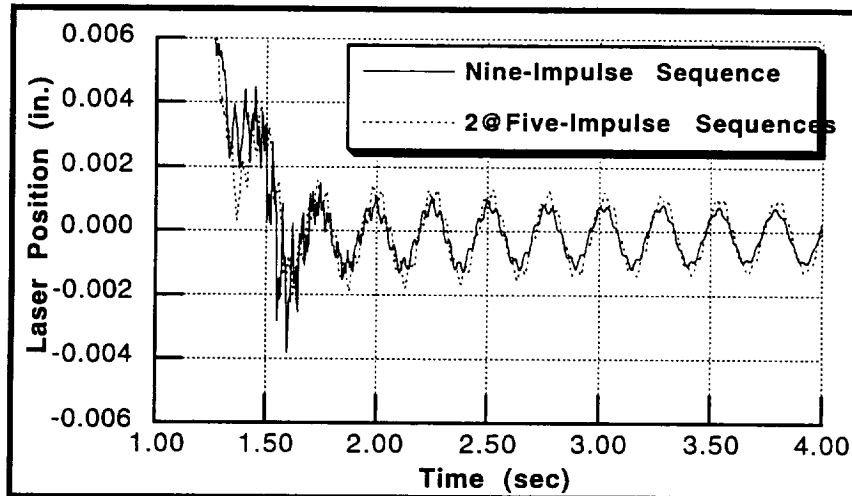


Figure 45: Close-Up of Shaped Residual Vibration.

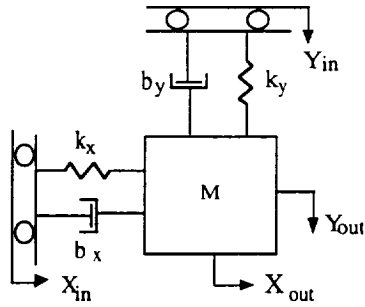


Figure 46: Two-Mode Model of a Flexible System Under PD Control.

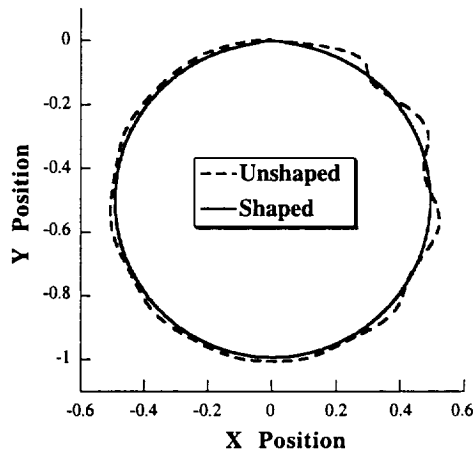


Figure 47: Comparison of ZVD Shaped and Unshaped Responses to a Unit-Circle Input ($f_1=f_2=1$ Hz, $z_1=z_2=0.05$, Unshaped Command Length=10 sec.).

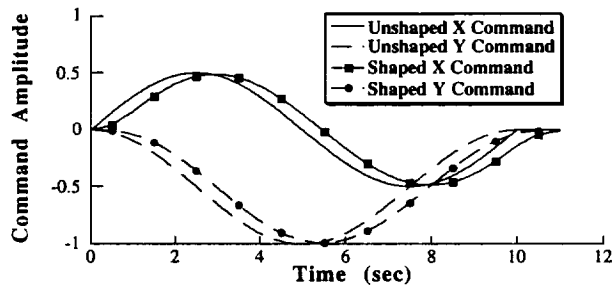


Figure 48: Unshaped and ZVD Shaped Command.

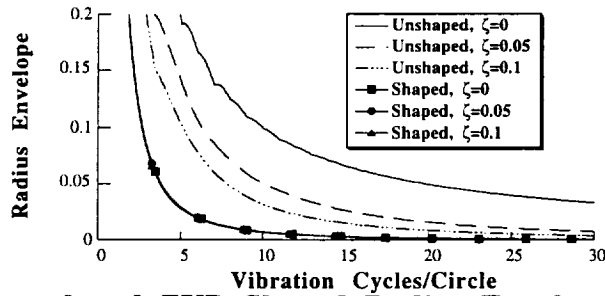


Figure 49: Unshaped and ZVD Shaped Radius Envelopes ($r = 1$). Note: All Shaped Curves are Essentially the Same.

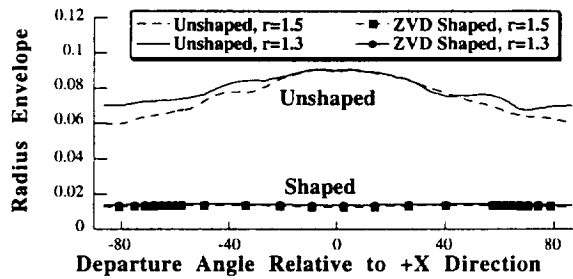


Figure 50: Radius Envelope as a Function of Departure Angle for Cycles/Circle=11.

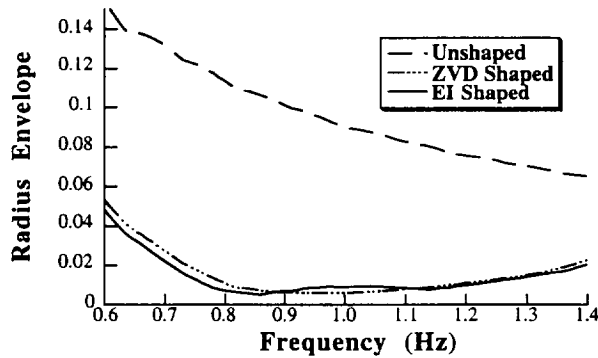


Figure 51: Response Radius Envelope as System Frequency Varies (Inputs Were Designed for $f_1=1$ Hz, $r=1$, $z_1=z_2=0$, and Cycles/Circle=11).

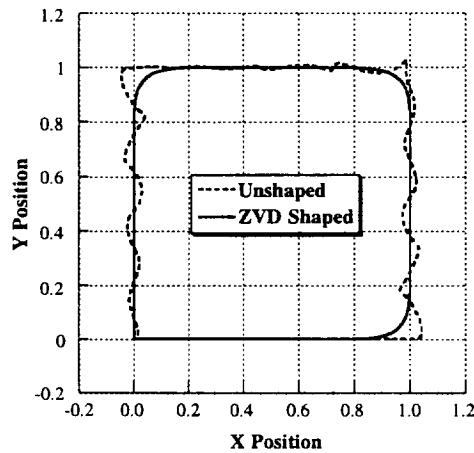


Figure 52: Comparison of ZVD Shaped and Unshaped Responses to a Unit-Square Input ($r = 1.5$, $z_1=z_2=0.05$, Cycles/Square=15).

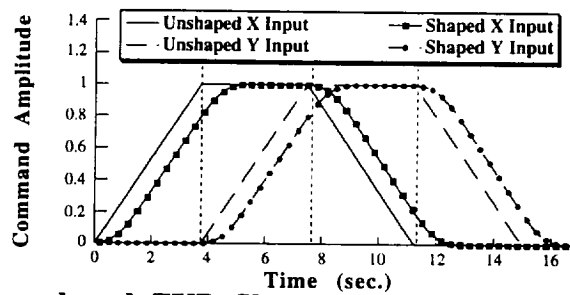


Figure 53: Unshaped and ZVD Shaped Commands for a Square Trajectory.

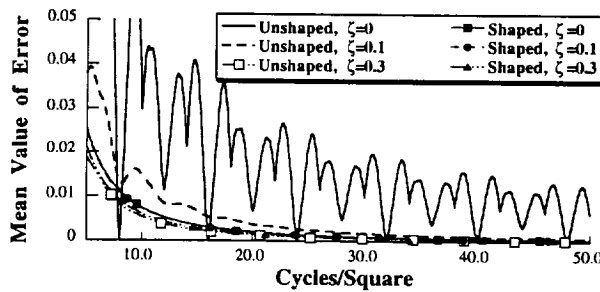


Figure 54: ZVD Shaped, EI Shaped, and Unshaped Mean Error as a Function of Cycles/Square and z . ($r = 1.5$).

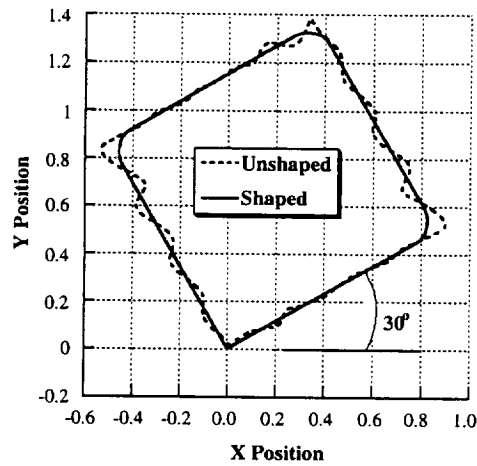


Figure 55: Unshaped and ZVD Shaped Responses to a Unit-Square Input with a Departure Angle of 30° Relative to the Low Mode ($r = 1.5$, $z_1=z_2=0.05$, Cycles/Square=15).

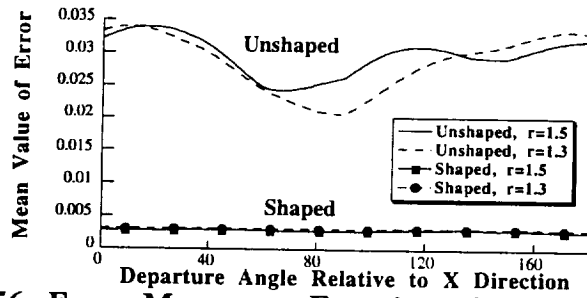


Figure 56: Error Mean as a Function of Departure Angle for Cycles/Square=17.

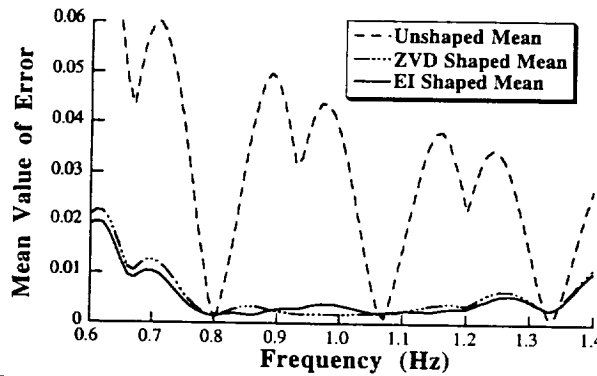


Figure 57: Mean of Square Tracking Error as System Frequency Varies (Inputs Were Designed for $f_1=1$ Hz, $r=1$, $z_1=z_2=0$, and Cycles/Square=15).

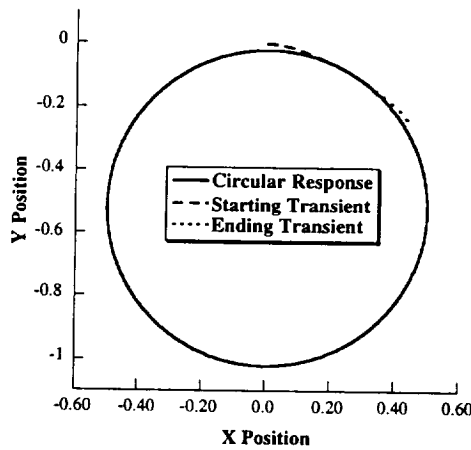


Figure 58: A Circular Response From Shaping a Larger than Desired Radius and Allowing for Transients ($r=1$, $z_1=z_2=0$, Cycles/Square=8).

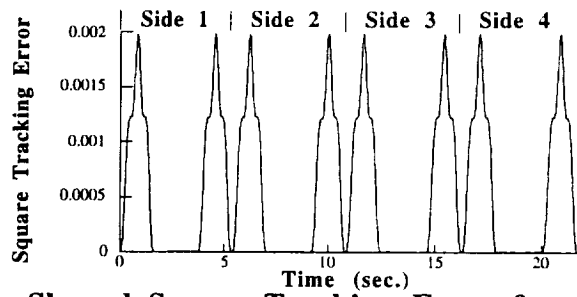


Figure 59: Shaped Square Tracking Error for $r=1.5$, $z_1=z_2=0$, Cycles/Square=15, and Departure Angle = 30° .

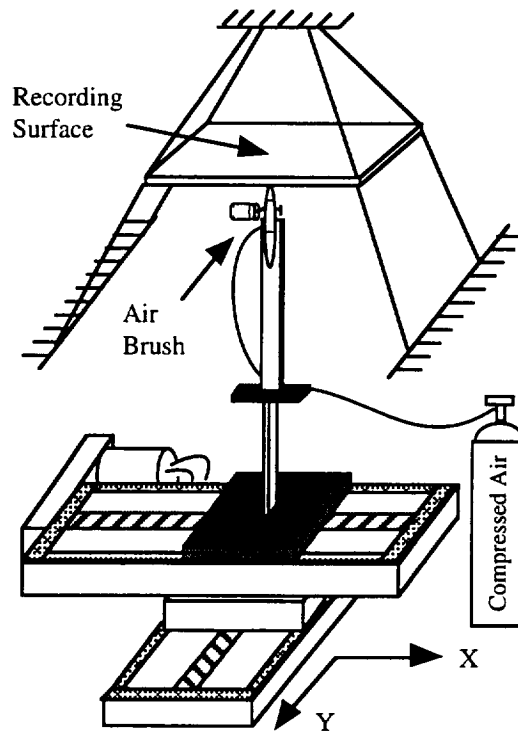


Figure 60: Experimental Setup

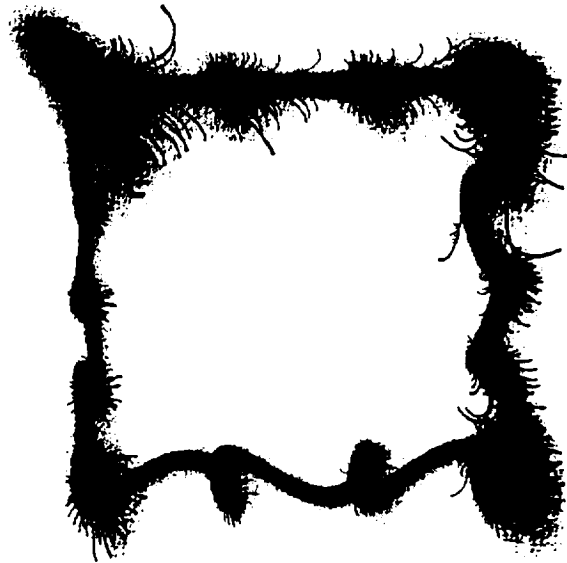


Figure 61: Unshaped 3x3 in. Square Trajectory

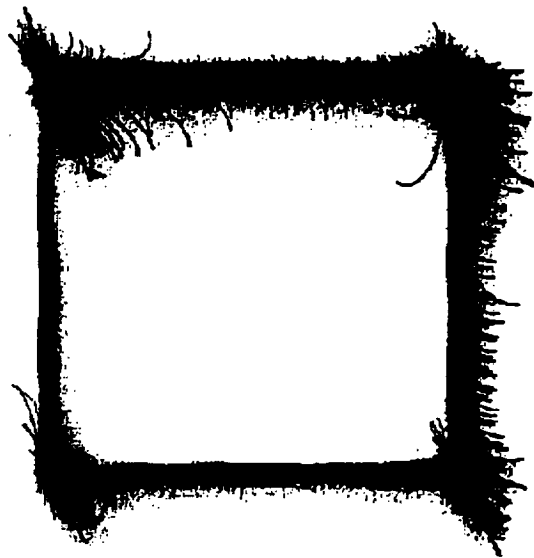


Figure 62: EI Shaped 3x3 in. Square Trajectory

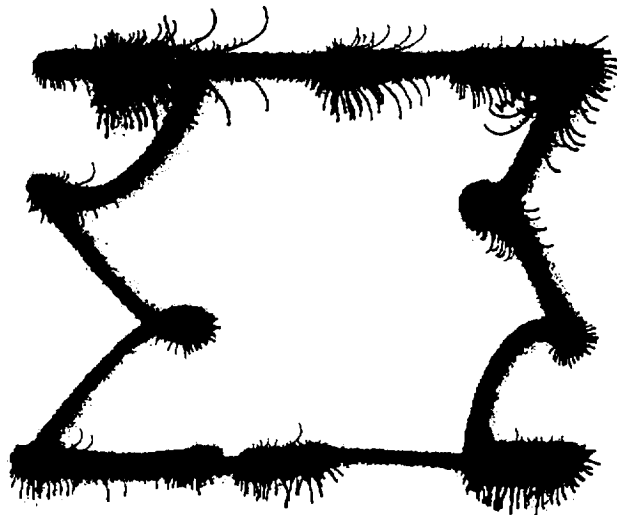


Figure 63: Unshaped 3x3 in. Square Trajectory with Added Endpoint Mass

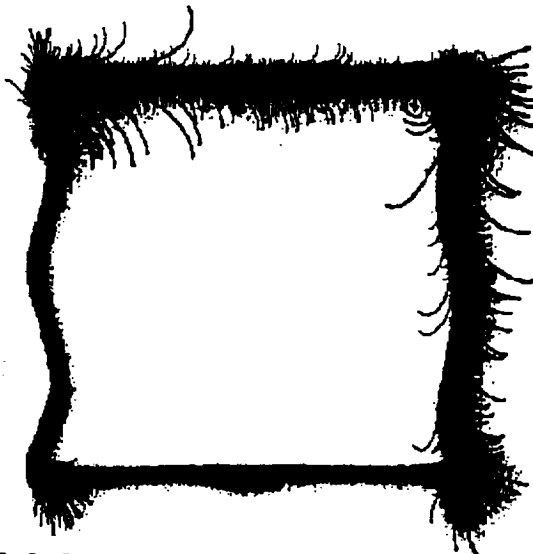


Figure 64: EI Shaped 3x3 in. Square Trajectory with Added Endpoint Mass

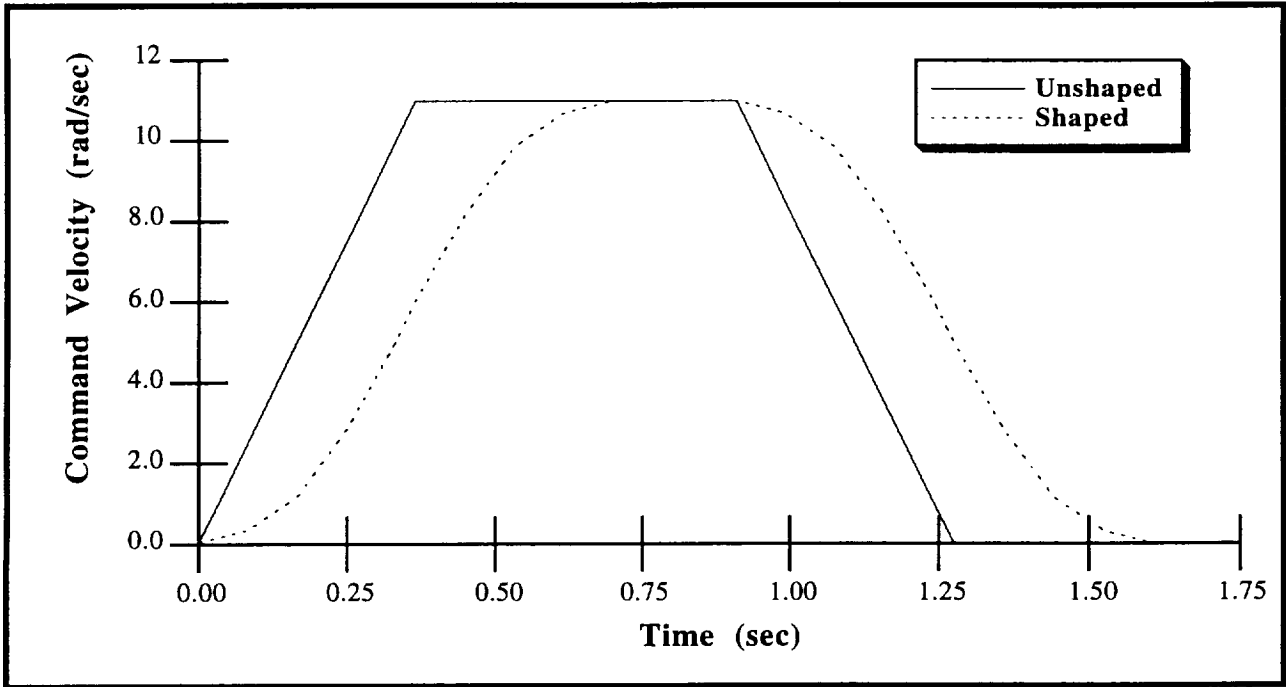


Figure 65: Shaped and Unshaped Trapezoidal Velocity Profiles. 10 rad. Slew, Shaping Sequence for 3.8 Hz. and 8.2 Hz., No Damping.

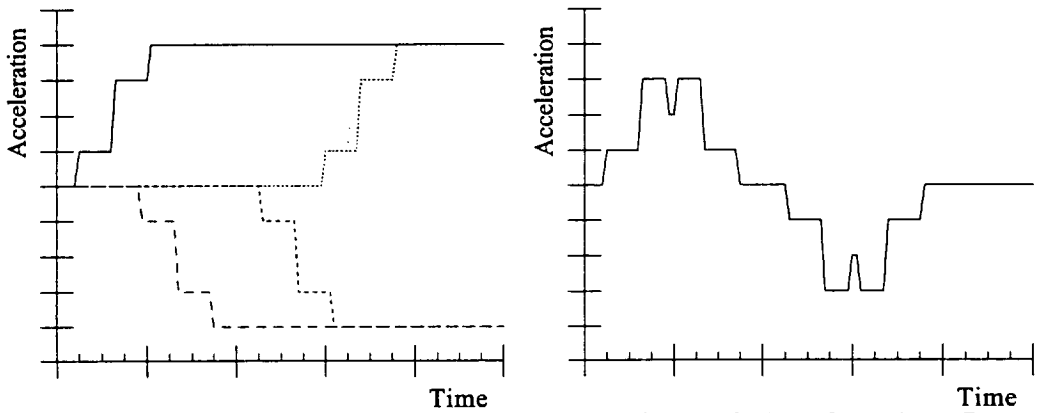


Figure 66 a, b: Superposition of Four Shaped Acceleration Steps Yields a Shaped Trapezoidal Velocity Profile

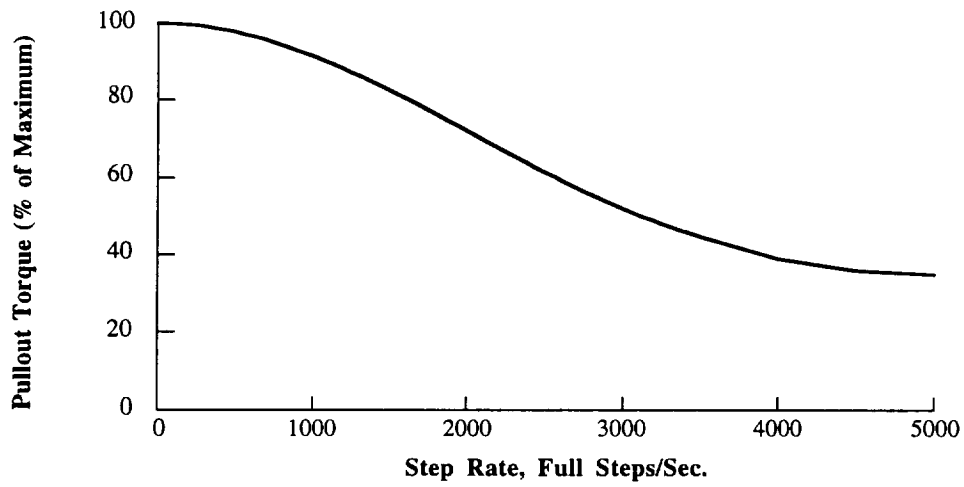


Figure 67: Pullout Torque-Speed Curve for a Stepper Motor.

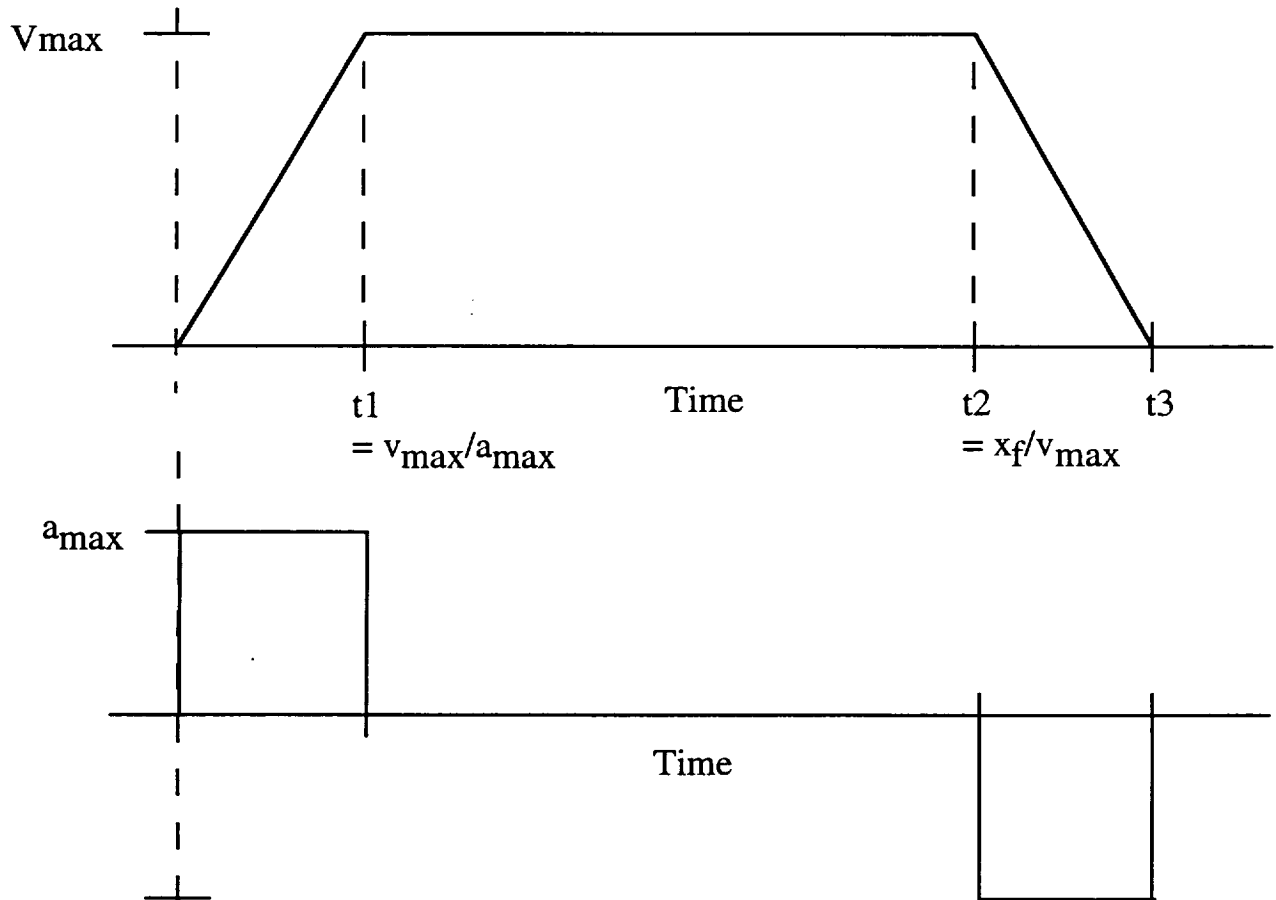


Figure 68 a, b: Trapezoidal Velocity Profile and Its Corresponding Acceleration Profile

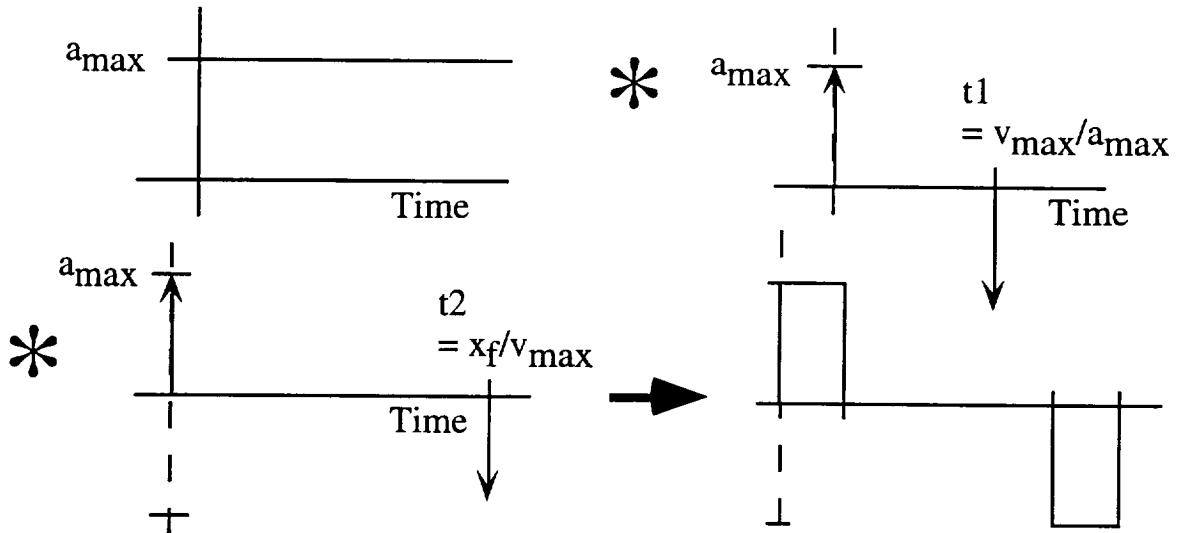


Figure 69 a,b,c,d: Convolution of a Step in Acceleration with Two Two-Impulse Shapers Yields the Acceleration Profile for a Trapezoidal Velocity Profile.

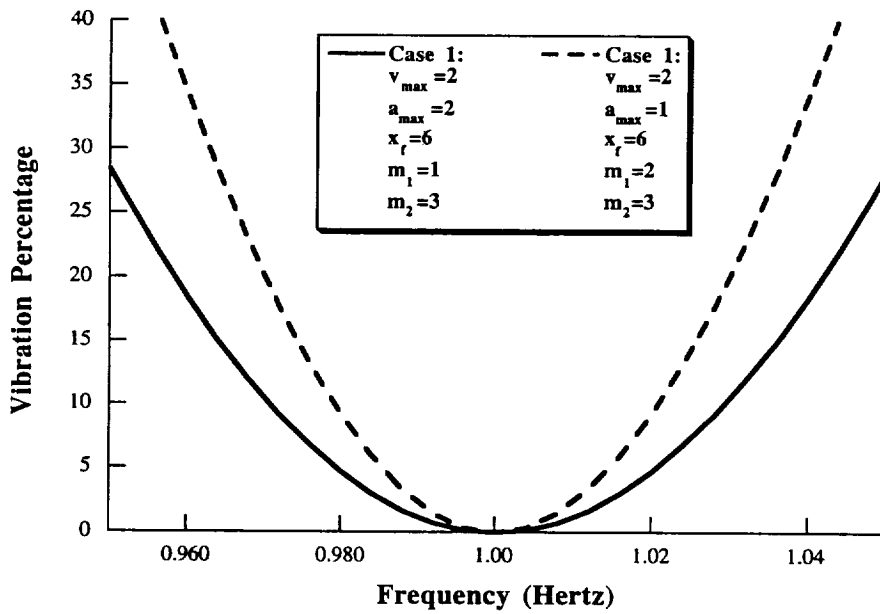


Figure 70: Insensitivity Degrades When m_1 is Increased.

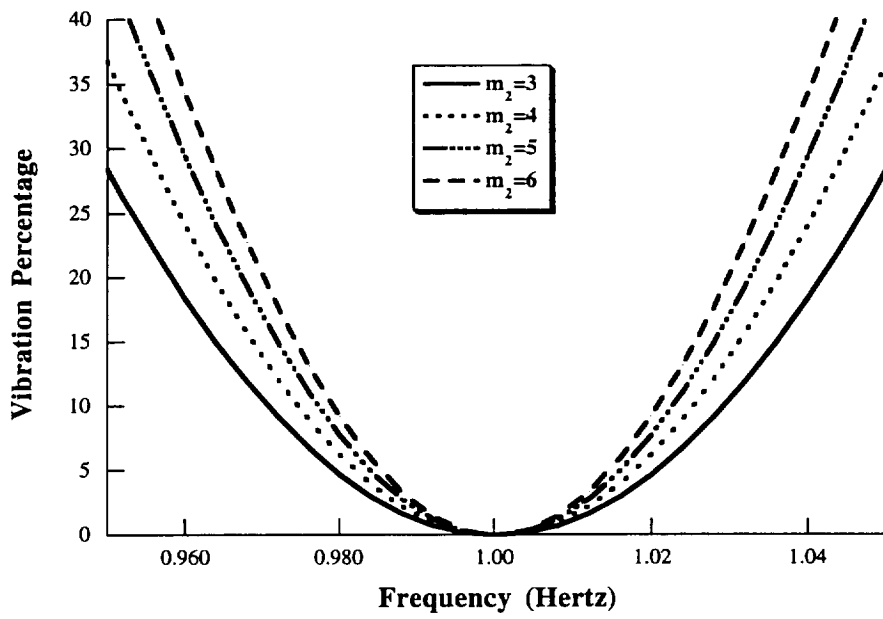


Figure 71: Insensitivity Degrades When m_2 is Increased.

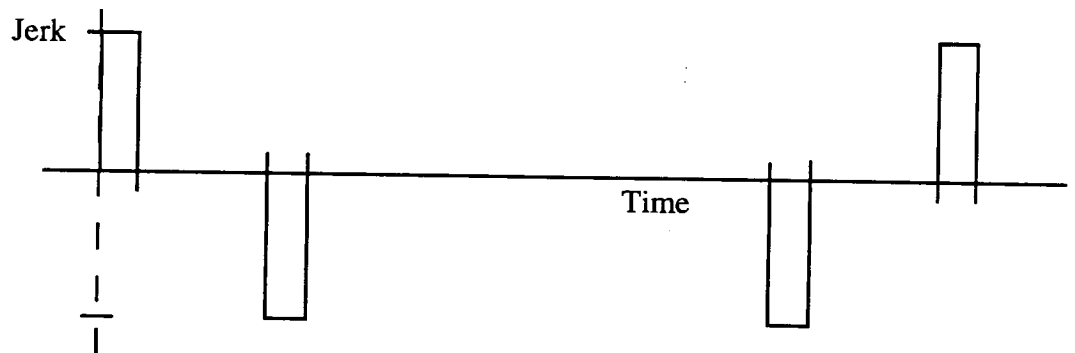
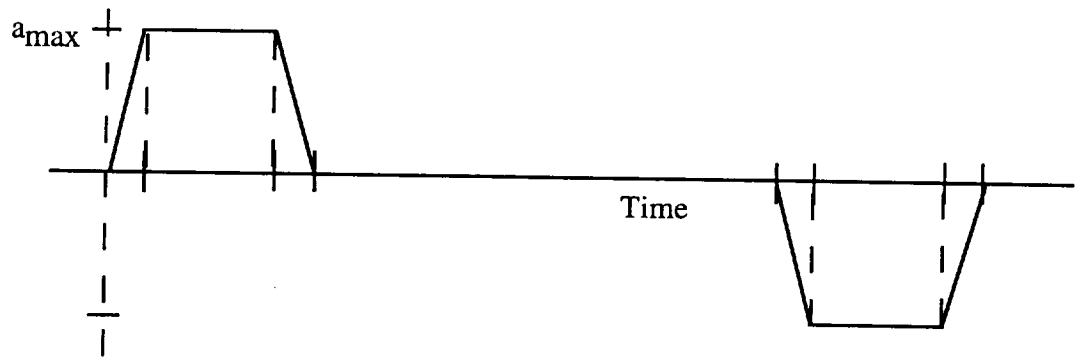
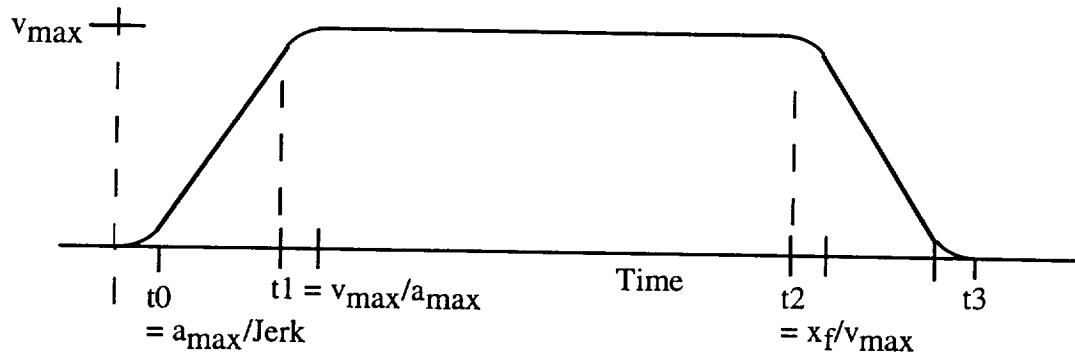


Figure 72 a,b,c: S-Curve Velocity Profile and its Corresponding Acceleration and Jerk Profiles.

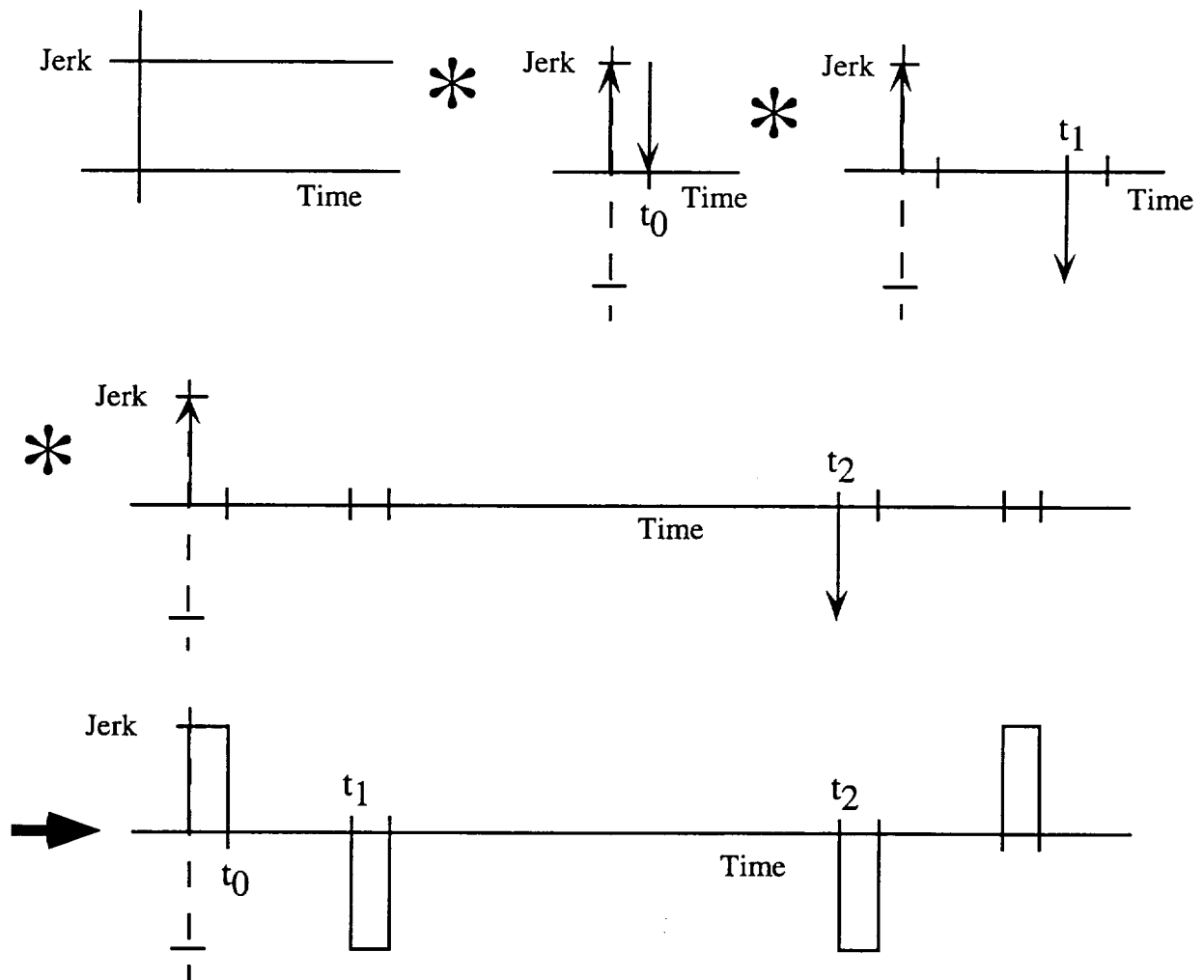


Figure 73 a,b,c,d,e: The Jerk Profile for the S-Curve Velocity Profile is Composed of a Step in Jerk Convolved with Three Two-Impulse Shapers.

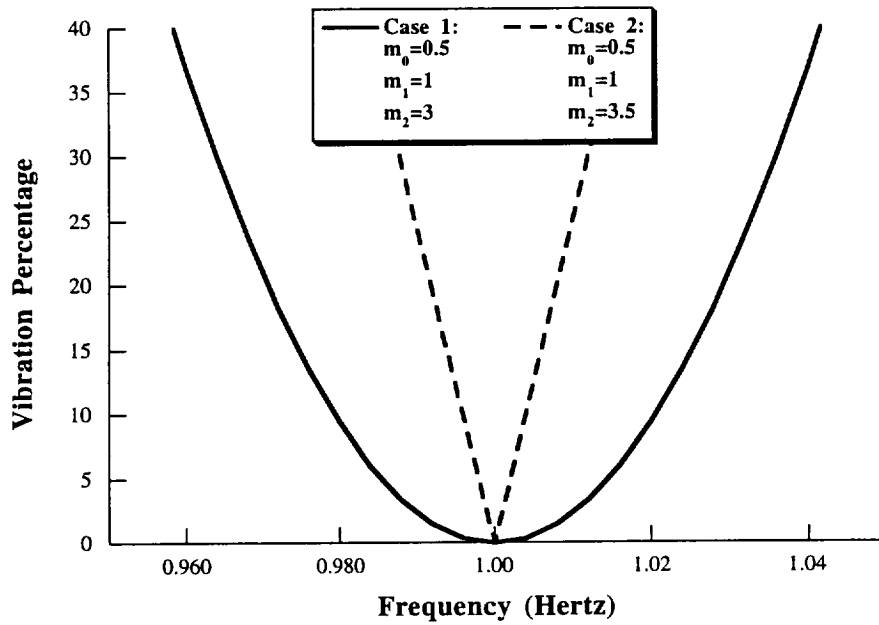


Figure 74: Insensitivity is Improved When More Than One S-Curve Metric is an Integer.

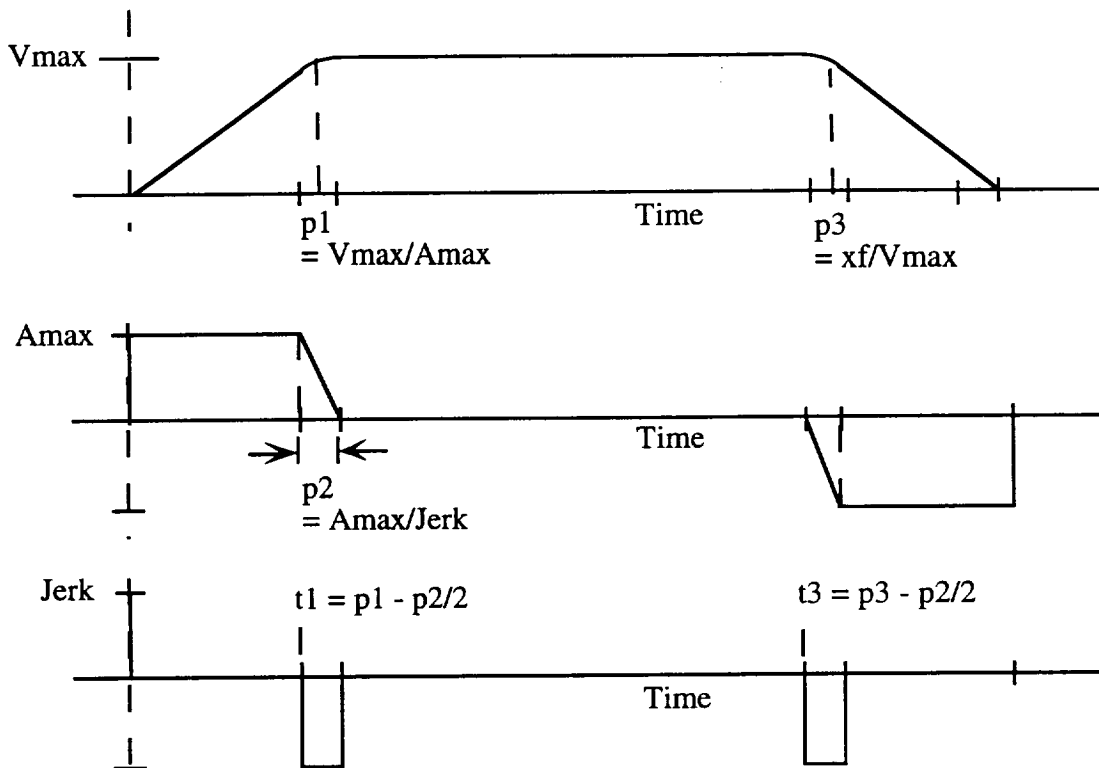


Figure 75 a,b,c: A Parabolic Velocity Profile and its Corresponding Acceleration and Jerk Profiles.

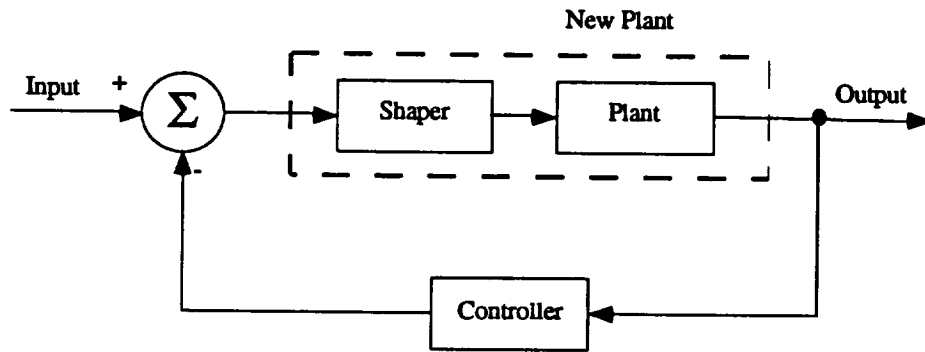


Figure 76: Block Diagram of a Shaper Inside of a Closed Loop (SIL)

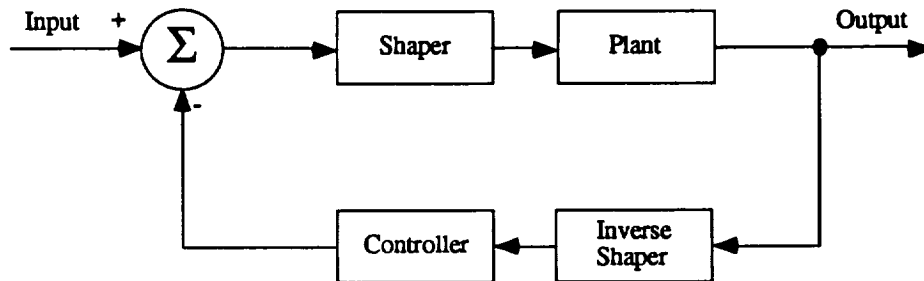


Figure 77: Block Diagram of a Shaper and Inverse Shaper Inside of a Closed Loop. This Block is functionally equivalent to Figure 78.

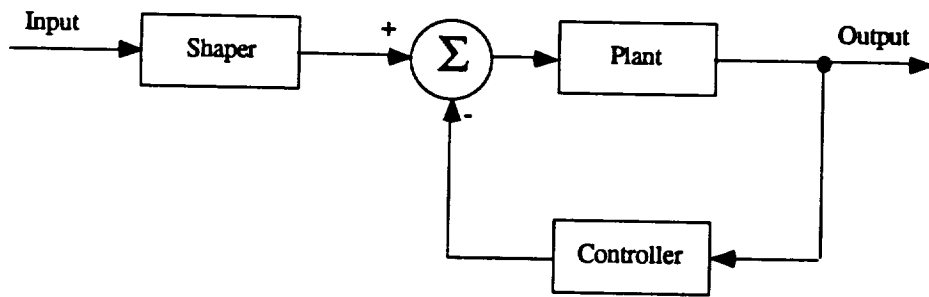


Figure 78: Block Diagram of a Conventional Shaped System.

GAMS Input Shaping™ Scripts

William Singhose
Convolve, Inc.
© 1993-95 All rights reserved.

Contents

I. Introduction.....	1
Road Map to Scripts.....	1
Overview of Script Structure.....	1
GAMS Formatting Conventions.....	1
Running GAMS.....	2
Procedure for Digital Scripts.....	2
II. Generic Input Shapers.....	4
ZV shapers.....	4
ZVD shapers.....	4
EI Shapers.....	4
Multiple-Hump EI Shapers.....	4
SI Shapers.....	4
Frequency Sampled Shapers.....	5
Least Squares Shapers.....	5
III. Negative Input Shapers.....	5
Negative ZV shapers.....	5
Negative ZVD shapers.....	5
Negative EI Shapers.....	5
Negative Multiple-Hump EI Shapers.....	5
Negative SI Shapers.....	6
IV. Constant-Amplitude Pulse Shapers.....	6
CAP ZV shapers.....	6
CAP ZVD shapers.....	6
CAP EI Shapers.....	6
CAP Multiple-Hump EI Shapers.....	7
References.....	7
Appendix A - Hard Copies of GAMS Scripts.....	8
Generic Input Shapers.....	8
ZV-DIG.gms.....	8
2-DIG.gms.....	9
EI.gms.....	10
EI-DIG.gms.....	12
SI0HUMP.gms.....	14
SI1HUMP.gms.....	17
IMP23TR.gms.....	21
HUMP01TR.gms.....	23
FS.gms.....	25
LP-2MOD.gms.....	27
Negative Input Shapers.....	30
NEGZV.gms.....	30
NEGZVD.gms.....	31
NEGEI.gms.....	32
Constant-Magnitude Shapers.....	35
CAPZV.gms.....	35
CVCAPZV.gms.....	36
CAPEI.gms.....	37

I. Introduction

GAMS (General Algebraic Modeling System) is a linear and nonlinear optimization program. The program is driven by user-generated scripts that describe the optimization problem. The scripts contain the constraint equations that are to be satisfied, the function to be minimized or maximized, resolution criteria, and input/output commands.

This manual describes GAMS scripts that were written to design input shapers. There are a large variety of scripts, each written to meet a distinct set of constraint equations. To help the user locate and run the appropriate script, this introduction will provide a road map for locating a script for the system under consideration. The procedure for running GAMS will then be reviewed.

Road Map to Scripts

The listing and description of scripts is broken down into the following subheadings:

- **Generic Input Shapers.**

These shaper are used for the majority of applications. They are used on systems with no special actuator limitations or abilities. If nothing is know about the system except its frequencies and damping ratios, then these are the shapers to be used.

These scripts will solve for ZV[3], ZVD[3], EI[7, 9], SI, and various types of multiple-mode shapers (this terminology is explained in the appropriate section). Scripts with the prefix "LP" solve for multiple shapers by looping through a given variable. For instance, many of the looping scripts find the desired input shaper over a large range of damping ratio, so that curve fit solutions as a function of damping can be obtained.

- **Negative Input Shapers.**

These shaper allow for faster rise times by using negative impulses in the shaper.

These shapers will lead to shaped inputs whose magnitude exceeds the unshaped input, so the system must be able to handle brief periods of over-currenting [2, 8].

- **Constant-Magnitude Shapers.**

These shapers are used on systems that are equipped with constant-magnitude actuators, such as on/off jets [5, 6].

Overview of Script Structure

The overall structure of an Input Shaping™ GAMS script is shown in Figure 1. The script shown in Figure 1 solves for a negative ZV shaper. The script has been broken down into 4 major sections: Parameters, Variables, Equations, and Output. Section 1 is where the parameters are declared and entered. These entries are constants that do not change during the course of the optimization. Calculations based on the parameter values are usually done in this section. Section 2 is where variables are declared. Initial guesses at the correct values of the variables can be entered here, as well as, bounds on the variable values. Section 3 declares and lists the constraint equations to be satisfied. The cost function to be minimized or maximized is also listed in this section. Section 4 controls the output from GAMS. Output files and formats are defined in this section.

GAMS Formatting Conventions

When entering information into GAMS scripts, certain formatting conventions must be observed. A few of the more useful, problematic and non-intuitive conventions will be mentioned here. For more information, see [1].

Comment lines are created by putting an asterisk at the start of the line. Tabs cannot be used to format a script. If indentation or spacing is desired, then the space character must be used.

When making an initial guess at a variable value, the variable name must be followed by a ".L" extension. Upper and lower bounds on variable values are set with the extensions ".UP" and ".LO".

The proper use of semicolons is important. Semicolons are used after the last entry in a declaration. For example, the headings SCALAR, SETS, PARAMETER, etc. are followed by a list of entries, a semicolon must be typed after the last entry. See Figure 1. Semicolons are also used after each equation that is used to calculate a parameter (Fig. 1, PARAMETER), and after each constraint equation (Fig. 1, EQUATIONS).

Running GAMS

Once a script is selected and the parameters are entered. The script is then executed with the command

>Gams *scriptname*

or,

>Gamsw *scriptname* (for running under Windows)

The output is located either in *scriptname .lst*, if a DISPLAY command is used, or in an output file specified near the bottom of the script, as was shown in Figure 1.

Errors that occur during the execution of a script are listed in the .LST file. The location of the errors is flagged by four asterisks, '****'.

Procedure for Digital Scripts

Most shapers can be designed in the continuous domain and then converted to a digital sequence by moving the impulses to the nearest digital time slot, or breaking each impulse into two impulses and placing them at the two closest digital time locations. When the sampling period is less than about 20 times the system's highest mode, then shapers should be designed with digital scripts. These scripts require the impulses to occur at the digital time step.

Designing a digital shaper is an iterative process. In addition to entering frequencies, damping ratios, and a sampling period, the designer must enter the number of impulses in the input shaper. By entering the sampling frequency and the number impulses, the designer has chosen the length of the shaper. If the script is executed and an optimal solution is found, then the length of the shaper should be reduced. This process is repeated until a script fails to find an optimal solution. If a script fails to find a solution, then the number of impulses must be increased until the minimum length shaper is determined.

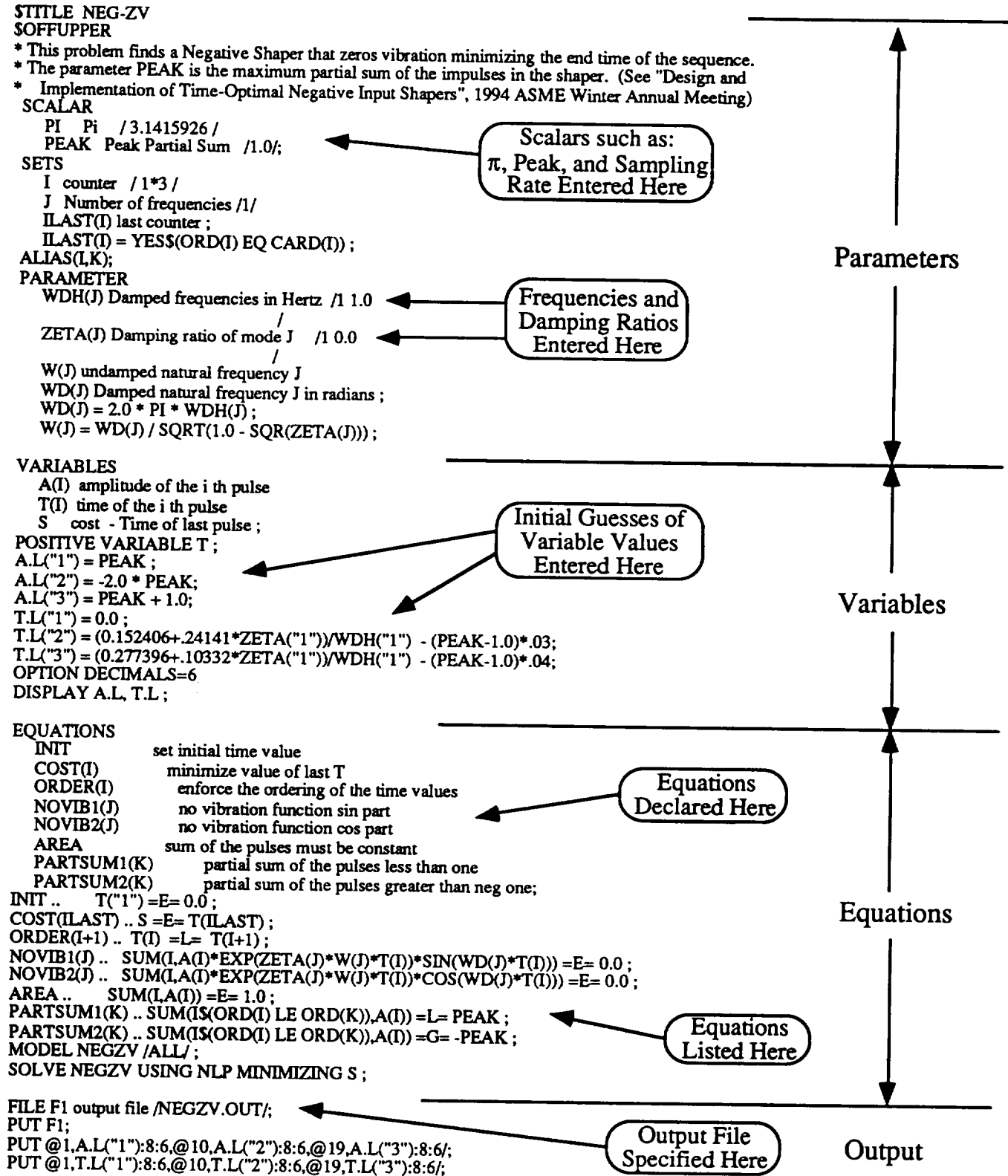


Figure 1: Structure of Input Shaping™ Scripts

II. Generic Input Shapers

ZV shapers

ZV shapers meet the constraints of Zero residual Vibration [3]. There is no constraint on insensitivity; consequently these shapers will work only if the frequencies are known very accurately and do not change during the course of operation. The scripts used to calculate ZV shapers are:

ZV-DIG.GMS*¹

ZVD shapers

ZVD shapers meet the constraints of Zero residual Vibration and zero Derivative of the residual vibration expression [3]. The scripts used to calculate ZV shapers are:

1-DIG.GMS – 1 mode digital version.

2-DIG.GMS – 2 mode digital version.*

3-DIG.GMS – 3 mode digital version.

Any of the above scripts can be converted to an n-mode script by changing the set counter to n, and entering n frequencies and damping ratios.

EI Shapers

EI shapers meet the extra-insensitive constraints [7, 9]. A small level of vibration, V, is allowed when the model is exact. V is measured as percentage of the unshaped vibration, i.e., V=0.05 means the residual vibration will be 5% of the unshaped level when the system model is exact. The vibration is then forced to zero at two frequencies, one higher than the modeling frequency and the other lower than the modeling frequency. This formulation leads to shapers that are significantly more insensitive to modeling errors than the ZVD shapers. The scripts that calculate EI shapers are:

EI.GMS*

LPEI.GMS

EI-DIG.GMS – digital 1 mode version*

EI-2M.GMS – 2 mode version (not debugged?)

Multiple-Hump EI Shapers

SI Shapers

SI shapers allow the insensitivity to be specified. The designer needs to know how many humps will be in the sensitivity curve for the desired insensitivity, so that the appropriate script can be chosen. There are equations that determine the number of humps given the insensitivity, or the scripts can be run in a trial and error manner. The scripts are:

SIOHUMP.GMS*

SI1HUMP.GMS*

SI2HUMP.GMS*

NEW2HUMP.GMS

There are codes that can be used to determine the transition from 2 to 3 impulses and the transition from 3 to 4 impulses. These codes are:

IMP23TR.GMS*

IMP34TR.GMS

The scripts that can find the approximate transitions between the number of humps are:

HUMP01TR.GMS*

HUMP12TR.GMS

HUMP23TR.GMS

¹ A printout for each code followed by an asterisk is included in Appendix A.

Frequency Sampled Shapers

Frequency sampled shapers are designed by requiring the vibration to be low at a number of specified frequencies [4]. For example, a frequency sampled shaper can be designed by requiring the vibration to be below a level V for 10 frequencies in the range from 0.8 Hz to 1.2 Hz. There are no constraints on derivatives in this formulation. This can be an effective brute force method for designed insensitivity over a specified range. Unfortunately, the insensitivity curve usually goes up between the sampled frequencies. This requires a fine spacing of the specified frequencies. The scripts that use frequency sampling are:

FS.GMS*

Least Squares Shapers

Least squares shapers are designed with constraints that minimize the sum of the squares of the residual vibration at a specified number of frequencies. The scripts that determine these shapers are:

LP-1-MOD.GMS

LP-2-MOD.GMS*

III. Negative Input Shapers**Negative ZV shapers**

Negative ZV shapers meet the constraints of Zero residual Vibration [2, 8]. These shapers are allowed to contain negative impulses. The maximum impulse amplitude is controlled by a constraint on the value of the partial sum of the impulses. That is, as the impulses are summed together one by one starting with the first two impulses, the total at each step of the summation cannot exceed the value PEAK. There is no constraint on insensitivity; consequently these shapers will work only if the frequencies are known very accurately and do not change during the course of operation. The scripts used to calculate negative ZV shapers are:

NEGZV.GMS*

LPNEGZV.GMS

Negative ZVD shapers

Negative ZVD shapers meet the constraints of Zero residual Vibration and zero Derivative of the residual vibration expression [2, 8]. These shapers are allowed to contain negative impulses. The maximum impulse amplitude is controlled by a constraint on the value of the partial sum of the impulses. That is, as the impulses are summed together one by one starting with the first two impulses, the total at each step of the summation cannot exceed the value PEAK. The scripts used to calculate negative ZVD shapers are:

NEGZVD.GMS*

LPNEGZVD.GMS

Negative EI Shapers

Negative EI shapers meet the extra-insensitive constraints [2, 8]. A small level of vibration, V , is allowed when the model is exact. V is measured as percentage of the unshaped vibration, i.e., $V=0.05$ means the residual vibration will be 5% of the unshaped level when the system model is exact. The vibration is then forced to zero at two frequencies, one higher than the modeling frequency and the other lower than the modeling frequency. This formulation leads to shapers that are significantly more insensitive to modeling errors than the ZVD shapers. The scripts that calculate negative EI shapers are:

NEG EI.GMS*

Negative Multiple-Hump EI Shapers

These scripts have not been written yet.

Negative SI Shapers

These scripts have not been written yet.

IV. Constant-Amplitude Pulse Shapers

Constant-Amplitude Pulse Shapers (CAP Shapers) must be convolved with step inputs [5, 6]. They generate a series of pulses, where the amplitude of the pulses is usually set equal to the maximum actuator effort. CAP shapers are used for two types of motions: rest-to-rest motions and accelerations to constant velocity. CAP shapers for rest-to-rest motions lead to an even number of bang-bang pulses. CAP shapers for acceleration to constant velocity can lead to an odd number of bang-bang pulses, or a command that consists only of positive valued pulses. Using only positive pulses is desirable in some cases because a system may not be equipped with equally balanced constant force actuators. Constant Velocity CAP shapers are distinguished with the prefix CV. For example, a ZV constant amplitude shaper for accelerating to a constant velocity would be referred to as a CVCAPZV shaper.

CAP ZV shapers

CAP ZV shapers meet the constraints of Zero residual Vibration [5, 6]. Additional constraints require the impulse amplitudes to be: [1, -2, 2, 1]. There is no constraint on insensitivity; consequently these shapers will work only if the frequencies are known very accurately and do not change during the course of operation. The desired move distance must be entered and an effective mass for the system. Distance and mass can easily be iterated to the correct ratio if tests on the system can be performed. The scripts used to calculate CAP ZV shapers are:

CAPZV.GMS*
 LPZV.GMS (Loops through a parameter such as move distance)
 CVCAPZV.GMS*

CAP ZVD shapers

CAP ZVD shapers meet the constraints of Zero residual Vibration and zero Derivative of the residual vibration expression [5, 6]. Additional constraints require the impulse amplitudes to be: [1, -2, 2, -2, 2, 1]. The desired move distance must be entered and an effective mass for the system. Distance and mass can easily be iterated to the correct ratio if tests on the system can be performed. The scripts used to calculate CAP ZV shapers are:

CAPZVD.GMS
 LPZVD.GMS (Loops through a parameter such as move distance)
 CVCAPZVD.GMS
 KRCVZVD.GMS

CAP EI Shapers

CAP EI shapers meet the extra-insensitive constraints [5, 6]. Additional constraints require the impulse amplitudes to be: [1, -2, 2, -2, 2, 1]. A small level of vibration, V, is allowed when the model is exact. V is measured as percentage of the unshaped vibration, i.e., V=0.05 means the residual vibration will be 5% of the unshaped level when the system model is exact. The vibration is then forced to zero at two frequencies, one higher than the modeling frequency and the other lower than the modeling frequency. This formulation leads to shapers that are significantly more insensitive to modeling errors than the ZVD shapers. The scripts that calculate CAP EI shapers are:

CAPEI.GMS*
 LPEI.GMS
 LPNEWEI.GMS
 CAPEI-2M.GMS
 LPEI-2M.GMS

EI-VARYD.GMS	(varies the move distance)
ROTARYEI.GMS	(used for the rotary table)
ROTEI-2M.GMS	(two-mode rotary table)
CVCAPEI.GMS	
4IMPEI.GMS	

CAP Multiple-Hump EI Shapers

CAP Multiple-Hump EI shapers satisfy multiple-hump extra-insensitive constraints [5]. The height of the humps in the sensitivity curve are set equal to a small level of vibration, V . The sensitivity curve is forced to zero at a frequency lower than the hump at the lowest frequency and higher than the highest frequency hump. In between the humps, the sensitivity curve is also forced to zero. Additional constraints require the impulse amplitudes to be: [1, -2, ... 2, 1]. The scripts that calculate CAP EI shapers are:

CV2HMPEI.GMS

References

- [1] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*. Redwood City, CA: The Scientific Press, 1988.
- [2] B.W. Rappole, N.C. Singer, and W.P. Seering, "Input Shaping With Negative Sequences for Reducing Vibrations in Flexible Structures," *American Control Conference*. San Francisco, CA, 1993.
- [3] N.C. Singer and W.P. Seering, "Preshaping Command Inputs to Reduce System Vibration," *ASME Journal of Dynamic Systems, Measurement, and Control*, March, 1990.
- [4] N.C. Singer and W.P. Seering, "An Extension of Command Shaping Methods for Controlling Residual Vibration Using Frequency Sampling," *Proceedings of the IEEE International Conference on Robotics and Automation*. Nice, France, 1992.
- [5] W. Singhose, S. Derezinski, and N. Singer, "Extra-Insensitive Shapers for Controlling Flexible Spacecraft," *AIAA Guidance, Navigation, and Control Conference*. Scottsdale, AZ, 1994.
- [6] W. Singhose, S. Derezinski, and N. Singer, "Input Shapers for Improving the Throughput of Torque-Limited Systems," *IEEE Conference on Control Applications*. Glasgow, Scotland, 1994.
- [7] W. Singhose, W. Seering, and N. Singer, "Residual Vibration Reduction Using Vector Diagrams to Generate Shaped Inputs," *ASME Journal of Mechanical Design*, June, 1994.
- [8] W. Singhose, N. Singer, and W. Seering, "Design and Implementation of Time-Optimal Negative Input Shapers," *ASME Winter Annual Meeting*. Chicago, IL, 1994.
- [9] W.E. Singhose, W.P. Seering, and N.C. Singer, "Shaping Inputs to Reduce Vibration: A Vector Diagram Approach," *IEEE Conference on Robotics and Automation*. Cincinnati, OH, 1990.

GAMS Scripts

Generic Input Shapers

ZV-DIG.gms

\$TITLE ZV-DIG.gms

\$OFFUPPER

* ZV - This problem finds a ZV shaper that zeros vibration and

* minimizes the first deriv. for n frequencies digital version

* copyright 1993 by Convolve, Inc.

*

OPTIONS DECIMALS=6

SCALAR

DT time spacing / .01 /

PI Pi / 3.1415926 /;

SETS

I counter / 1*10/

J Number of frequencies / 1*4/;

PARAMETER

T(I) time of the i th pulse

WDH(J) Damped frequencies in Hertz

/ 1 10.

2 17.0

3 27.0

4 42.5/

ZETA(J) Damping ratio of mode J

/ 1 .05

2 .0

3 .0

4 .0/

W(J) undamped natural frequency J

WD(J) Damped natural frequency J in radians ;

WD(J) = 2.0 * PI * WDH(J) ;

W(J) = WD(J) / SQRT(1.0 - SQR(ZETA(J))) ;

T(I) = DT * (ORD(I) - 1);

VARIABLES

A(I) amplitude of the i th pulse

S cost - deriv.

SS(J) cost - deriv. sin part

SC(J) cost - deriv. cos part ;

POSITIVE VARIABLE A ;

EQUATIONS

COST

derivative cost function

PDERIV1(J) first derivative function sin part pos

PDERIV2(J) first derivative function cos part pos

NDERIV1(J) first derivative function sin part neg

NDERIV2(J) first derivative function cos part neg

NOVIB1(J) no vibration function sin part

NOVIB2(J) no vibration function cos part

AREA sum of the pulses must be constant;

PDERIV1(J) .. -SS(J) =G= -1.0e6*SUM(I,A(I)*T(I)*EXP(ZETA(J)*W(J)*T(I))*
SIN(WD(J)*T(I)));PDERIV2(J) .. -SC(J) =G= -1.0e6*SUM(I,A(I)*T(I)*EXP(ZETA(J)*W(J)*T(I))*
COS(WD(J)*T(I)));

```

NDERIV1(J) .. -SS(J) =G= 1.0e6*SUM(I,A(I)*T(I)*EXP(ZETA(J)*W(J)*T(I))*
              SIN(WD(J)*T(I)));
NDERIV2(J) .. -SC(J) =G= 1.0e6*SUM(I,A(I)*T(I)*EXP(ZETA(J)*W(J)*T(I))*
              COS(WD(J)*T(I)));
COST ..      S =E= -SUM(J,SS(J) + SC(J)) ;
NOVIB1(J) .. SUM(I,A(I)*EXP(ZETA(J)*W(J)*T(I))*SIN(WD(J)*T(I))) =E= 0.0 ;
NOVIB2(J) .. SUM(I,A(I)*EXP(ZETA(J)*W(J)*T(I))*COS(WD(J)*T(I))) =E= 0.0 ;
AREA ..     SUM(I,A(I)) =E= 1.0 ;

```

MODEL DAMP /ALL/ ;

SOLVE DAMP USING LP MINIMIZING S ;

DISPLAY A.L, A.M ;

2-DIG.gms

\$TITLE 2-DIG.gms

\$OFFUPPER

* This problem finds a sequence that zeros vibration and first deriv.

* minimizing the second deriv. for 2 frequencies digital version

* copyright 1990 by Convolve, Inc.

*

OPTIONS DECIMALS=6

SCALAR

DT time spacing /01 /

PI Pi /3.1415926 / ;

SETS

I counter / 1*20 /

J Number of frequencies /1,2/;

PARAMETER

T(I) time of the i th pulse

WDH(J) Damped frequencies in Hertz

/1 6.5
2 17 /

ZETA(J) Damping ratio of mode J

/1 .1
2 .0/

W(J) undamped natural frequency J

WD(J) Damped natural frequency J in radians ;

WD(J) = 2.0 * PI * WDH(J) ;

W(J) = WD(J) / SQRT(1.0 - SQR(ZETA(J))) ;

T(I) = DT * (ORD(I) - 1);

VARIABLES

A(I) amplitude of the i th pulse

S cost - second deriv.

SS(J) cost - second deriv. sin part

SC(J) cost - second deriv. cos part ;

POSITIVE VARIABLE A ;

EQUATIONS

COST second derivative cost function

PSECDER1(J) second derivative function sin part pos

PSECDER2(J) second derivative function cos part pos

NSECDER1(J) second derivative function sin part neg

NSECDER2(J) second derivative function cos part neg
 DERIV1(J) first derivative function sin part
 DERIV2(J) first derivative function cos part
 NOVIB1(J) no vibration function sin part
 NOVIB2(J) no vibration function cos part
 AREA sum of the pulses must be constant;

```
PSECDER1(J) .. -SS(J) =G= -1.0e6*SUM(I,A(I)*SQR(T(I))*
EXP(ZETA(J)*W(J)*T(I))*SIN(WD(J)*T(I))) ;
PSECDER2(J) .. -SC(J) =G= -1.0e6*SUM(I,A(I)*SQR(T(I))*
EXP(ZETA(J)*W(J)*T(I))*COS(WD(J)*T(I))) ;
NSECDER1(J) .. -SS(J) =G= 1.0e6*SUM(I,A(I)*SQR(T(I))*
EXP(ZETA(J)*W(J)*T(I))*SIN(WD(J)*T(I))) ;
NSECDER2(J) .. -SC(J) =G= 1.0e6*SUM(I,A(I)*SQR(T(I))*
EXP(ZETA(J)*W(J)*T(I))*COS(WD(J)*T(I))) ;
COST .. S =E= -SUM(J,SS(J) + SC(J)) ;
DERIV1(J) .. 1000.0*SUM(I,A(I)*T(I)*EXP(ZETA(J)*W(J)*T(I))*
SIN(WD(J)*T(I))) =E= 0.0 ;
DERIV2(J) .. 1000.0*SUM(I,A(I)*T(I)*EXP(ZETA(J)*W(J)*T(I))*
COS(WD(J)*T(I))) =E= 0.0 ;
NOVIB1(J) .. SUM(I,A(I)*EXP(ZETA(J)*W(J)*T(I))*SIN(WD(J)*T(I))) =E= 0.0 ;
NOVIB2(J) .. SUM(I,A(I)*EXP(ZETA(J)*W(J)*T(I))*COS(WD(J)*T(I))) =E= 0.0 ;
AREA .. SUM(I,A(I)) =E= 1.0 ;
```

MODEL DAMP /ALL/ ;

SOLVE DAMP USING LP MINIMIZING S ;

DISPLAY A.L, A.M ;

EI.gms

\$TITLE EI.gms

\$OFFUPPER

- * This program finds an extra-insensitive shaper for 1 mode.
- * It finds the original solution when there is no damping.
- * In the presence of damping, the symmetry of the touchdown frequencies is sacrificed - giving skewed insensitivity.
- * To avoid a large amount of skew, the distance between the first touchdown and the modeling frequency is maximized.
- * This keeps the curve from skewing all the way to the right.

SCALAR

WDH Damped frequency in Hertz /1/
 ZETA Damping ratio /0.0/
 PI Pi /3.1415927 /
 VLIM Vibration limit at WDH /0.05/;

SETS

I counter / 1*3 /
 L Number of touchdown frequencies /1*2/;

PARAMETER

W undamped natural frequency in radians
 WD Damped natural frequency in radians
 DELTAW approx distance to touchdown freq. from W;
 WD = 2.0 * PI * WDH ;
 W = WD / SQRT(1.0 - SQR(ZETA)) ;

* The following equation is exact for no damping. The $(1-ZETA)$ term is a hack for damping.

$$\text{DELTAW} = (\text{W}/\text{PI}) * \text{ARCTAN}(2 * \text{SQRT}(\text{VLIM}) / (1 - \text{VLIM})) / (1 - \text{ZETA});$$

VARIABLES

A(I) amplitude of the i th pulse
 T(I) time of the i th pulse
 TW(L) touchdown frequencies (undamped)
 WP FREQUENCY OF VIBRATION PEAK
 S cost - distance from TW("2") to TW("1") - maximize this
 COSSUM SUM AIEXPCOS
 SINSUM SUM AIEXPSIN
 TZCOSSUM SUM AI*TI*ZETA*EXP*COS
 TZSINSUM SUM AI*TI*ZETA*EXP*SIN
 TSQRTZCOS SUM AI*TI*SQRT1-Z2*EXP*COS
 TSQRTZSIN SUM AI*TI*SQRT1-Z2*EXP*SIN

POSITIVE VARIABLE T ;
 POSITIVE VARIABLE TW ;
 POSITIVE VARIABLE WP ;
 POSITIVE VARIABLE A ;

* LIMITS ON THE TOUCHDOWN FREQUENCIES *****

TW.UP("1") = W;
 TW.LO("2") = W;

* INITIAL GUESSES *****

TW.L("1") = W-DELTAW;
 TW.L("2") = W+DELTAW;
 WP.L = 1.0*W ;
 T.L("1") = 0.0 ;
 T.L("2") = .5/WDH;
 T.L("3") = 1.0/WDH;
 A.L("1") = 1.0/
 (1+2*(1-VLIM)*EXP(-ZETA*W*T.L("2"))/(1+VLIM) + EXP(-2*ZETA*W*T.L("3")));
 A.L("2") = 2*(1-VLIM)/(1+VLIM)*EXP(-ZETA*W*T.L("2"))/
 (1+2*(1-VLIM)*EXP(-ZETA*W*T.L("2"))/(1+VLIM) + EXP(-2*ZETA*W*T.L("3")));
 A.L("3") = EXP(-2*ZETA*W*T.L("3"))/
 (1+2*(1-VLIM)*EXP(-ZETA*W*T.L("2"))/(1+VLIM) + EXP(-2*ZETA*W*T.L("3")));
 COSSUM.L = VLIM;
 SINSUM.L = 0;
 TZCOSSUM.L = 0;
 TZSINSUM.L = 0;
 TSQRTZCOS.L = (.5/WDH)*((1.0-VLIM)/2)*COS(W*.5/WDH) +
 (1.0/WDH)*((1.0+VLIM)/4)*COS(W*1.0/WDH);
 TSQRTZSIN.L = 0;

OPTION DECIMALS=6

DISPLAY A.L, T.L, TW.L,TSQRTZCOS.L, DELTAW ;

EQUATIONS

INIT set initial time value
 FINAL set final time value
 COST maximize TW("2")-TW("1")
 ORDER(I) enforce the ordering of the time values
 VIBLIMIT set vibration limit at the freq. of interest
 AREA sum of the pulses must be constant
 DERIV deriv constraint
 WPLW INSURES WP IS LESS THAN W
 ECOSSUM SUM AIEXPCOS

```

ESINSUM          SUM AIEXPSIN
ETZCOSSUM       SUM AI*TI*ZETA*EXP*COS
ETZSINSUM       SUM AI*TI*ZETA*EXP*SIN
ETSQRTZCOS      SUM AI*TI*SQRT1-Z2*EXP*COS
ETSQRTZSIN      SUM AI*TI*SQRT1-Z2*EXP*SIN
TOUCH1(L)       touchdown of the insensitivity curve
TOUCH2(L)       touchdown of the insensitivity curve;

INIT ..         T("1") =E= 0.0 ;
FINAL ..        T("3") =E= 1.0/WDH ;
COST ..         S =E= TW("2") - TW("1") ;
ORDER(I+1) ..  T(I) =L= T(I+1) ;
WPLW ..         WP =L= W ;
ECOSSUM ..      COSSUM =E= SUM(I,A(I)*EXP(ZETA*WP*T(I))*COS(WP*SQRT(1.0 -
SQR(ZETA))*T(I))) ;
ESINSUM ..      SINSUM =E= SUM(I,A(I)*EXP(ZETA*WP*T(I))*SIN(WP*SQRT(1.0 -
SQR(ZETA))*T(I))) ;
ETZCOSSUM ..    TZCOSSUM =E=
SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP*T(I))*COS(WP*SQRT(1.0 - SQR(ZETA))*T(I))) ;
ETZSINSUM ..    TZSINSUM =E= SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP*T(I))*SIN(WP*SQRT(1.0
- SQR(ZETA))*T(I))) ;
ETSQRTZCOS ..   TSQRTZCOS =E= SUM(I,A(I)*T(I)*SQRT(1-
SQR(ZETA))*EXP(ZETA*WP*T(I))*COS(WP*SQRT(1.0 - SQR(ZETA))*T(I))) ;
ETSQRTZSIN ..   TSQRTZSIN =E= SUM(I,A(I)*T(I)*SQRT(1-
SQR(ZETA))*EXP(ZETA*WP*T(I))*SIN(WP*SQRT(1.0 - SQR(ZETA))*T(I))) ;
VIBLIMIT ..     EXP(-ZETA*WP*T("3"))*SQRT(
    SQR(SINSUM) + SQR(COSSUM)) =L= VLIM ;
DERIV ..        (-ZETA*T("3")*(SQR(COSSUM)+SQR(SINSUM)) + SINSUM*(TSQRTZCOS +
    TZSINSUM) + COSSUM*(TZCOSSUM - TSQRTZSIN))/
    (EXP(ZETA*WP*T("3"))*SQRT(SQR(COSSUM)+SQR(SINSUM))) =E= 0.0 ;
AREA ..         SUM(I,A(I)) =E= 1.0 ;
TOUCH1(L) ..    SUM(I,A(I)*EXP(ZETA*TW(L)*T(I))*
    SIN(TW(L)*SQRT(1.0 - SQR(ZETA))*T(I))) =E= 0.0 ;
TOUCH2(L) ..    SUM(I,A(I)*EXP(ZETA*TW(L)*T(I))*
    COS(TW(L)*SQRT(1.0 - SQR(ZETA))*T(I))) =E= 0.0 ;

MODEL DAMP /ALL/ ;

SOLVE DAMP USING NLP MAXIMIZING S ;

DISPLAY "Extra-Insensitive", A.L, A.M ,T.L, T.M, TW.L ;

```

EI-DIG.gms

\$TITLE EI-DIG.gms

\$OFFUPPER

- * This program finds a digital EI shaper for 1 mode.
- * In the presence of damping, the symmetry of the touchdown frequencies is sacrificed - giving skewed insensitivity.
- * (The difference between the two distances from the modeling frequency to the touchdown points is limited to 1% of W.)
- * To avoid a large amount of skew, the distance between the first touchdown and the modeling frequency is maximized.
- * This keeps the curve from skewing all the way to the right.
- * To ensure max vibration occurs at W, and not somewhere near W,
- * the vibration at $W\Delta = 0.999*W$ is kept below VLIM.

SCALAR

WDH Damped frequency in Hertz /18.5/
 ZETA Damping ratio /0.05/

```

PI  Pi  / 3.1415926 /
DT  time spacing      / .01 /
VLIM  Vibration Limit at WDH      / .03 /;

```

SETS

```

I  counter / 1*7 /
L  Number of touchdown frequencies /1*2/
ILAST(I) last counter ;
ILAST(I) = YES$(ORD(I) EQ CARD(I)) ;

```

PARAMETER

```

T(I) time of the i th pulse
W  undamped natural frequency
WD  Damped natural frequency in radians
DELTAW approx distance to touchdown freq. from W;
WD = 2.0 * PI * WDH ;
W = WD / SQRT(1.0 - SQR(ZETA)) ;
* The following equation is exact for no damping. The ((1-ZETA)) term is a hack for damping.
DELTAW = (W/PI)*ARCTAN(2*SQRT(VLIM)/(1-VLIM))/(1-ZETA);
T(I) = DT * (ORD(I) - 1);

```

VARIABLES

```

A(I) amplitude of the i th pulse
TW(L) touchdown frequencies (undamped)
WP  FREQUENCY OF VIBRATION PEAK
S  cost - distance from TW("2") to TW("1") - maximize this
COSSUM      SUM AIEXPCOS
SINSUM      SUM AIEXPSIN
TZCOSSUM    SUM AI*TI*ZETA*EXP* COS
TZSINSUM    SUM AI*TI*ZETA*EXP* SIN
TSQRTZCOS   SUM AI*TI*SQRT1-Z2*EXP* COS
TSQRTZSIN   SUM AI*TI*SQRT1-Z2*EXP* SIN

```

```

POSITIVE VARIABLE TW ;
POSITIVE VARIABLE WP ;
POSITIVE VARIABLE A ;

```

* LIMITS ON THE TOUCHDOWN FREQUENCIES *****

```

TW.UP("1") = W;
TW.LO("2") = W;

```

* INITIAL GUESSES *****

```

TW.L("1") = W-DELTAW;
TW.L("2") = W+DELTAW;
WP.L = 1.0*W ;
A.L("1") = 1.0/
(1+2*(1-VLIM)*EXP(-ZETA*W*T("4"))/(1+VLIM) + EXP(-2*ZETA*W*T("7")));
A.L("4") = 2*(1-VLIM)/(1+VLIM)*EXP(-ZETA*W*T("4"))/
(1+2*(1-VLIM)*EXP(-ZETA*W*T("4"))/(1+VLIM) + EXP(-2*ZETA*W*T("7")));
A.L("7") = EXP(-2*ZETA*W*T.L("7"))/
(1+2*(1-VLIM)*EXP(-ZETA*W*T("4"))/(1+VLIM) + EXP(-2*ZETA*W*T("7")));
COSSUM.L = VLIM;
SINSUM.L = 0;
TZCOSSUM.L = 0;
TZSINSUM.L = 0;
TSQRTZCOS.L = (.5/WDH)*((1.0-VLIM)/2)*COS(W*.5/WDH) +
(1.0/WDH)*((1.0+VLIM)/4)*COS(W*1.0/WDH);

```

TSQRTZSIN.L = 0;

OPTION DECIMALS=6

EQUATIONS

TOUCH1(L)	touchdown of the insensitivity curve
TOUCH2(L)	touchdown of the insensitivity curve
COST	maximize TW("2")-TW("1")
VIBLIMIT	set vibration limit at the freq. of interest
AREA	sum of the pulses must be constant
DERIV	deriv constraint
WPLW	INSURES WP IS LESS THAN W
ECOSSUM	SUM AIEXPCOS
ESINSUM	SUM AIEXPSIN
ETZCOSSUM	SUM AI*TI*ZETA*EXP* COS
ETZSINSUM	SUM AI*TI*ZETA*EXP* SIN
ETSQRTZCOS	SUM AI*TI*SQRT1-Z2*EXP* COS
ETSQRTZSIN	SUM AI*TI*SQRT1-Z2*EXP* SIN ;

```

TOUCH1(L) .. SUM(I,A(I)*EXP(ZETA*TW(L)*T(I))*
                SIN(TW(L)*SQRT(1.0 - SQR(ZETA))*T(I))) =E= 0.0 ;
TOUCH2(L) .. SUM(I,A(I)*EXP(ZETA*TW(L)*T(I))*
                COS(TW(L)*SQRT(1.0 - SQR(ZETA))*T(I))) =E= 0.0 ;
COST ..      S =E= TW("2") - TW("1") ;
WPLW ..     WP =L= W ;
ECOSSUM ..  COSSUM =E= SUM(I,A(I)*EXP(ZETA*WP*T(I))*COS(WP*SQRT(1.0 -
SQR(ZETA))*T(I))) ;
ESINSUM ..  SINSUM =E= SUM(I,A(I)*EXP(ZETA*WP*T(I))*SIN(WP*SQRT(1.0 -
SQR(ZETA))*T(I))) ;
ETZCOSSUM .. TZCOSSUM =E=
SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP*T(I))*COS(WP*SQRT(1.0 - SQR(ZETA))*T(I))) ;
ETZSINSUM .. TZSINSUM =E= SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP*T(I))*SIN(WP*SQRT(1.0
- SQR(ZETA))*T(I))) ;
ETSQRTZCOS .. TSQRTZCOS =E= SUM(I,A(I)*T(I)*SQRT(1-
SQR(ZETA))*EXP(ZETA*WP*T(I))*COS(WP*SQRT(1.0 - SQR(ZETA))*T(I))) ;
ETSQRTZSIN .. TSQRTZSIN =E= SUM(I,A(I)*T(I)*SQRT(1-
SQR(ZETA))*EXP(ZETA*WP*T(I))*SIN(WP*SQRT(1.0 - SQR(ZETA))*T(I))) ;
VIBLIMIT .. EXP(-ZETA*WP*T("7"))*SQRT(SQR(SINSUM) + SQR(COSSUM)) =L= VLIM ;
DERIV ..    (-ZETA*T("7"))*(SQR(COSSUM)+SQR(SINSUM)) + SINSUM*(TSQRTZCOS +
                TZSINSUM) + COSSUM*(TZCOSSUM - TSQRTZSIN))/
                (EXP(ZETA*WP*T("7"))*SQRT(SQR(COSSUM)+SQR(SINSUM))) =E= 0.0 ;
AREA ..    SUM(I,A(I)) =E= 1.0 ;

```

MODEL DAMP /ALL/ ;

SOLVE DAMP USING NLP MAXIMIZING S ;

DISPLAY "EI Shaper",A.L,TW.L ;

SI0HUMP.gms

\$TITLE SI0HUMP

\$OFFUPPER

* THIS PROGRAM SOLVES FOR THE SI SHAPER IN THE REGION
 * OF INSENSITIVITY WITH 0 HUMPS IN THE SENSITIVITY CURVE.
 * TURN OFF MOST STUFF PRINTED TO .LST FILE

\$OFFSYMREF \$OFFSYMLIST \$OFFUELLIST \$OFFUELXREF

OPTION SOLPRINT = OFF;

OPTION LIMROW = 0;

OPTION LIMCOL = 0;

SCALAR

WDH Damped frequency in Hertz /1.0/
 Z Damping ratio /0.0/
 PI Pi /3.1415926/
 SLV TO SOLVE OR NOT TO SOLVE /0.0/
 ZSTART STARTING VALUE OF Z /0.0/
 V Vibration Limit /0.1/
 INS STARTING INSENSITIVITY /0.15/
 IN INSENSITIVITY(INCREMENTS) /0.15/;

SETS

I counter /1*3/
 N Number of LIMITED frequencies /1*2/
 VCT LOOP THRU V /1*1/
 ICT LOOP THRU INSENSITIVITY /1*14/
 ZCT variable to loop thru Z /1*1/
 ILAST(I) LAST COUNTER;
 ILAST(I) = YES\$(ORD(I) EQ CARD(I));

PARAMETER

WP(N) LIMITED FREQUENCIES IN RADIANs
 WD Damped natural frequency in radians;
 WD = 2.0 * PI * WDH ;

VARIABLES

A(I) amplitude of the i th pulse
 T(I) time of the ith pulse
 S cost -TIME OF LAST IMPULSE;

POSITIVE VARIABLE A ;
 POSITIVE VARIABLE T ;
 T.UP(I) = 1.1;
 A.UP(I) = 1.0;
 OPTION DECIMALS=7

EQUATIONS

COST(I) MINIMIZE T(ILAST)
 ORDER(I) ENFORCE TIME ORDER OF IMPULSES
 INITIAL SET T("1") =0
 VIBLIMIT(N,I) Set vibration limit at the freqs. of interest
 AREA Sum of the pulses must be 1;

ORDER(I+1) .. T(I) =L= T(I+1);
 INITIAL .. T("1") =E= 0.0;
 COST(ILAST) .. S =E= T(ILAST) ;
 VIBLIMIT(N,ILAST) .. $\text{EXP}(-Z*WP(N)*T(ILAST))*\text{SQRT}(\text{SQR}(\text{SUM}(I,A(I)*\text{EXP}(Z*WP(N)*T(I))*\text{SIN}(WP(N)*\text{SQRT}(1.0 - \text{SQR}(Z))*T(I)))) + \text{SQR}(\text{SUM}(I,A(I)*\text{EXP}(Z*WP(N)*T(I))*\text{COS}(WP(N)*\text{SQRT}(1.0 - \text{SQR}(Z))*T(I)))) =L= V ;$
 AREA .. SUM(I,A(I)) =E= 1.0 ;

MODEL SIOHUMP /ALL/ ;

FILE F1 output file /SIOHUMP.AMP/;
 FILE F2 output file /SIOHUMP.TME/;


```

LOOP(VCT,
* ---- RESET GUESSES ----
  A.L("1") = 0.5;
  A.L("2") = 0.5;
  T.L("2") = 0.25;
  T.L("3") = 0.5;
IN=INS;
LOOP(ICT,
  Z=ZSTART;
  LOOP(ZCT,
    WP("1") = (WD/SQRT(1.0-SQR(Z)))*(1.0-(0.5*IN));
    WP("2") = (WD/SQRT(1.0-SQR(Z)))*(1.0+(0.5*IN));

*      ---- Check to see if the insensitivity is small enough so that
*      a hump does not get within the insensitivity window.
*      If not, skip because constraints are not appropriate.----

    IF ((.0521806+4.93891*V-44.8588*V**2+330.228*V**3
      -1015.24*V**4+.0842162*Z+7.19696*V*Z-74.4955*V**2*Z
      +541.214*V**3*Z-1687.39*V**4*Z+.050139*Z**2
      +3.53268*V*Z**2-33.7336*V**2*Z**2+308.012*V**3*Z**2
      -655.988*V**4*Z**2+.204829*Z**3+11.9838*V*Z**3
      -159.96*V**2*Z**3+1042.21*V**3*Z**3
      -4433.49*V**4*Z**3 -IN -.001) GE 0.0,
      SLV=1;
    );

    IF (SLV EQ 1,
      SOLVE SIOHUMP USING NLP MINIMIZING S ;

*      --- Check for an unnecessary third impulse ----
      IF ((T.L("3") - T.L("2")) LE .000001,
        A.L("2") = A.L("2") + A.L("3");
        A.L("3") = 0.0;
        T.L("3") = 0.0;
      );
      IF (A.L("2") LE .000001,
        A.L("2") = A.L("3");
        A.L("3") = 0.0;
        T.L("2") = T.L("3");
        T.L("3") = 0.0;
      );
      PUT F1;
      PUT @1,V:4:2,@6,IN:6:4,@13,Z:4:2,@18,A.L("1"):9:7,@28,
        A.L("2"):9:7,@38,A.L("3"):9:7/;
      IF (SIOHUMP.MODELSTAT NE 2,
        PUT 'PUKE'/;
      );
      PUT F2;
      PUT @1,V:4:2,@6,IN:6:4,@13,Z:4:2,@18,T.L("2"):9:7,@28,
        T.L("3"):9:7/;
      IF (SIOHUMP.MODELSTAT NE 2,
        PUT 'PUKE'/;
      );
    );
  );
  );
SLV = 0;
Z=Z+0.01;
);

```

```
IN=IN+.005;
);
V=V+.01;
);
```

SI1HUMP.gms

\$TITLE SI1HUMP

\$OFFUPPER

* THIS PROGRAM SOLVES FOR THE SI SHAPER IN THE REGION
 * OF INSENSITIVITY WITH 1 HUMP IN THE SENSITIVITY CURVE.
 * TURN OFF PRINTING OF MAPS AND SOLVE STATEMENT IN .LST FILE
 \$OFFSYMREF OFFSYMLIST OFFUELLIST OFFUELXREF

OPTION SOLPRINT = OFF;
 OPTION LIMROW = 0;
 OPTION LIMCOL = 0;

SCALAR

WDH Damped frequency in Hertz /1.0/
 Z Damping ratio /0.0/
 PI Pi /3.1415926/
 V Vibration Limit /0.03/
 SLV TO SOLVE OR NOT TO SOLVE /0.0/
 AL1 STORES A1 FOR Z=0 /0.2575/
 AL2 STORES A2 FOR Z=0 /0.485/
 AL3 STORES A3 FOR Z=0 /0.485/
 AL4 STORES A4 FOR Z=0 /0.2575/
 TL2 STORES T2 FOR Z=0 /0.439/
 TL3 STORES T3 FOR Z=0 /0.875/
 TL4 STORES T4 FOR Z=0 /1.314/
 WP2 STORES WP("2") FOR Z=0 /1.10/
 WTD1 STORES WTD("1") FOR Z=0 /0.86/
 WTD2 STORES WTD("2") FOR Z=0 /1.14/
 OPTSOLN OPT SLN FOUND FOR THIS IN /0.0/
 INS STARTING INSENSITIVITY /0.18/
 IN INSENSITIVITY(INCREMENTS) /0.18/;

SETS

I counter /1*4/
 N Number of LIMITED frequencies /1*3/
 M NUMBER OF FREQS TO FREQ SAMPLE /1*9/
 P NUMBER OF TOUCHDOWN FREQS /1*2/
 VCT LOOP THRU V /1*1/
 ZCT variable to loop thru Z /1*31/
 ICT LOOP THRU INSENSITIVITY /1*120/
 ILAST(I) LAST COUNTER;
 ILAST(I) = YES\$(ORD(I) EQ CARD(I));

PARAMETER

WFS(M) FREQUENCY SAMPLING POINTS
 WD Damped natural frequency in radians;
 WD = 2.0 * PI * WDH ;

VARIABLES

WP(N) LIMITED FREQUENCIES IN RADIANS
 WTD(P) TOUCHDOWN FREQS FOR Z=0 CASE
 A(I) amplitude of the i th pulse
 T(I) time of the ith pulse
 S cost - TIME OF LAST IMPULSE;

POSITIVE VARIABLE A ;
 POSITIVE VARIABLE T ;
 POSITIVE VARIABLE WP;
 POSITIVE VARIABLE WTD;

WP.UP("1") = 1.0*WD;
 WP.UP("2") = 1.5*WD;
 WP.UP("3") = 2.0*WD;
 WTD.UP("1") = 1.5*WD;
 WTD.UP("2") = 2.0*WD;

A.UP(I) = 1.0;
 T.UP(I) = 1.5/WDH;

OPTION DECIMALS=7

EQUATIONS

COST(I) MINIMIZE T(ILAST)
 ORDER(I) ENFORCE TIME ORDER OF IMPULSES
 ORDERWP(N) ENFORCE THE ORDER OF WPS
 WTD1LWP2 WTD("1") IS LESS THAN WP("2")
 WTD2GWP2 WTD("2") IS GREATER THAN WP("2")
 INITIAL SET T("1") =0
 SYM ENFORCE AMPLITUDE SYM FOR Z=ZERO
 VIBLIMIT(N,I) Set vibration limit at the freqs. of interest
 FSVIBLIM(M,I) FREQUENCY SAMPLE SO ALT SOLN NOT POSSIBLE
 WTDVIBLIM(P,I)
 LVIBLIM(N,I) LIMIT VIBRATION AT WP3 BUT NOT NEC EQUAL TO V
 AREA Sum of the pulses must be 1
 ZERODERIV(N,I) Zero Derivative constraint;

ORDER(I+1) .. T(I) =L= T(I+1);
 ORDERWP(N+1) .. WP(N) =L= WP(N+1);
 INITIAL .. T("1") =E= 0.0;
 COST(ILAST) .. S =E= T(ILAST) ;
 SYM\$(Z EQ 0.0) .. A("1") =E= A("4"); .
 WTD1LWP2\$(Z EQ 0.0) .. WTD("1") =L= 0.98*WP("2");
 WTD2GWP2\$(Z EQ 0.0) .. WTD("2") =G= 1.02*WP("2");
 VIBLIMIT(N,ILAST)\$((ORD(N) EQ 1) OR (ORD(N) EQ 2))
 .. EXP(-Z*WP(N)*T(ILAST))*SQRT(SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*
 SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I)))) + SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*
 COS(WP(N)*SQRT(1.0 - SQR(Z))*T(I)))))) =E= V ;
 FSVIBLIM(M,ILAST)
 .. EXP(-Z*WFS(M)*T(ILAST))*SQRT(SQR(SUM(I,A(I)*EXP(Z*WFS(M)*T(I))*
 SIN(WFS(M)*SQRT(1.0 - SQR(Z))*T(I)))) + SQR(SUM(I,A(I)*EXP(Z*WFS(M)*T(I))*
 COS(WFS(M)*SQRT(1.0 - SQR(Z))*T(I)))))) =L= V ;
 WTDVIBLIM(P,ILAST)\$ (Z EQ 0.0)
 .. EXP(-Z*WTD(P)*T(ILAST))*SQRT(SQR(SUM(I,A(I)*EXP(Z*WTD(P)*T(I))*
 SIN(WTD(P)*SQRT(1.0 - SQR(Z))*T(I)))) + SQR(SUM(I,A(I)*EXP(Z*WTD(P)*T(I))*
 COS(WTD(P)*SQRT(1.0 - SQR(Z))*T(I)))))) =L= 0.001 ;
 LVIBLIM(N,ILAST)\$ (ORD(N) EQ 3)
 .. EXP(-Z*WP(N)*T(ILAST))*SQRT(SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*
 SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I)))) + SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*
 COS(WP(N)*SQRT(1.0 - SQR(Z))*T(I)))))) =L= V ;
 AREA .. SUM(I,A(I)) =E= 1.0 ;
 ZERODERIV(N,ILAST)\$ (ORD(N) EQ 2)

```

.. -EXP(-T(ILAST)*WP(N)*Z)*{
Z*T(ILAST)*SQRT(SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 -
SQR(Z))*T(I))))
+ SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I))))
+ { SUM(I,A(I)*EXP(Z*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 - SQR(Z))*T(I))
*SUM(I,A(I)*T(I)*EXP(T(I)*WP(N)*Z)*(SQRT(1-SQR(Z))*SIN(T(I)*WP(N)*SQRT(1-SQR(Z)))
-Z*COS(T(I)*WP(N)*SQRT(1-SQR(Z))))
- SUM(I,A(I)*EXP(Z*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I))
*SUM(I,A(I)*T(I)*EXP(T(I)*WP(N)*Z)*(SQRT(1-SQR(Z))*COS(T(I)*WP(N)*SQRT(1-SQR(Z)))
+Z*SIN(T(I)*WP(N)*SQRT(1-SQR(Z)))) }
/(SQRT(SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 -SQR(Z))*T(I))))
+ SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 -SQR(Z))*T(I)))) )
=E= 0.0;

```

```

MODEL S11HUMP /ALL/ ;
FILE F1 output file /S11HUMP.AMP/;
FILE F2 output file /S11HUMP.TME/;

```

```

LOOP(VCT,

```

```

    IN=INS;

```

```

* --- Guesses for starting at the beginning of a V ----

```

```

    WTD1 = WD*(1.0-(0.1*IN));

```

```

    WTD2 = WD*(0.1+IN);

```

```

    WP2 = WD*(1.0 + (0.48*IN));

```

```

* --- GUESSES FOR STARTING IN THE MIDDLE OF A V ---

```

```

*   WTD1 = 0.78*WD;

```

```

*   WTD2 = 1.13*WD;

```

```

*   WP2 = 0.94*WD;

```

```

    LOOP(ICT,

```

```

        WPL("2") = WP2;

```

```

        WTDL("1") = WTD1;

```

```

        WTDL("2") = WTD2;

```

```

* --- Base guesses for 3 or 4 impulse sequence on IMP4 ---

```

```

* --- SET A3 EQUAL TO ZERO AND A1=A4 TO SATISFY SYM EQN ---

```

```

    A.L("1") = AL1;

```

```

    A.L("2") = AL2;

```

```

    A.L("3") = AL3;

```

```

    A.L("4") = AL4;

```

```

    T.L("2") = TL2+.001;

```

```

    T.L("3") = TL3;

```

```

    T.L("4") = TL4+.002;

```

```

Z=0.0;

```

```

LOOP(ZCT,

```

```

* --- FIX WP1 AND WP3 FOR GIVEN INSENSITIVITY ---

```

```

    WP.FX("1") = (WD/SQRT(1.0-SQR(Z)))*(1.0-(0.5*IN));

```

```

    WP.FX("3") = (WD/SQRT(1.0-SQR(Z)))*(1.0+(0.5*IN));

```

```

* --- SET FREQUENCY SAMPLING POINTS ---

```

```

    WFS("1") = 1.01*WPL("1");

```

```

    WFS("2") = (WD/SQRT(1.0-SQR(Z)))*(1.0-(0.3*IN));

```

```

    WFS("3") = (WD/SQRT(1.0-SQR(Z)))*(1.0-(0.2*IN));

```

```

    WFS("4") = (WD/SQRT(1.0-SQR(Z)))*(1.0-(0.1*IN));

```

```

    WFS("5") = WD/SQRT(1.0-SQR(Z));

```

```

    WFS("6") = (WD/SQRT(1.0-SQR(Z)))*(1.0+(0.1*IN));

```

```

    WFS("7") = (WD/SQRT(1.0-SQR(Z)))*(1.0+(0.2*IN));

```

```

    WFS("8") = (WD/SQRT(1.0-SQR(Z)))*(1.0+(0.3*IN));

```

WFS("9") = 0.99*WP.L("3");

```

*
* ---- Check to see if there is THE CORRECT AMOUNT OF insensitivity to
* get 1 hump within the insensitivity window.
* If not skip because constraints are not appropriate. ----
* If not skip because constraints are not appropriate. ----
IF ((0.251664 + 11.3123*V- 132.714*V**2 + 1008.64*V**3
-3151.01*V**4+0.380913*Z+14.1447*V*Z-196.386*V**2*Z
+1605.83*V**3*Z-5286.74*V**4*Z + 0.43879*Z**2
+9.90331*V*Z**2-221.957*V**2*Z**2+1115.89*V**3*Z**2
-1742.17*V**4*Z**2+0.703582*Z**3 - 22.5705*V*Z**3
+349.461*V**2*Z**3-418.679*V**3*Z**3-4556.15*V**4*Z**3
-IN - .002) GE 0.0,
IF (IN - (.0521806+4.93891*V-44.8588*V**2+330.228*V**3
-1015.24*V**4+.0842162*Z+7.19696*V*Z-74.4955*V**2*Z
+541.214*V**3*Z-1687.39*V**4*Z+.050139*Z**2
+3.53268*V*Z**2-33.7336*V**2*Z**2+308.012*V**3*Z**2
-655.988*V**4*Z**2+.204829*Z**3+11.9838*V*Z**3
-159.96*V**2*Z**3+1042.21*V**3*Z**3
-4433.49*V**4*Z**3+.001) GE 0.0,
    SLV=1;
);
);

* ---- IF INSENSITIVITY IS APPROPRIATE - SOLVE ----
IF (SLV GE 0.9,
    SOLVE S11HUMP USING NLP MINIMIZING S ;

* ---- STORE VALUES OF 1ST OPTIMAL SOLN FOR GUESSES AT NEXT I ----
IF ((OPTSOLN EQ 0.0) AND (S11HUMP.MODELSTAT EQ 2),
    OPTSOLN = 1.0;
    AL1=A.L("1");
    AL2=A.L("2");
    AL3=A.L("3");
    AL4=A.L("4");
    TL2=T.L("2");
    TL3=T.L("3");
    TL4=T.L("4");
    WP2=WP.L("2");
    WTD1=WTD.L("1");
    WTD2=WTD.L("2");
);

* ---- Check for an unnecessary fourth impulse ----
IF ((T.L("4") - T.L("3")) LE .0000002,
    IF(A.L("4") GE 0.0000002,
        A.L("3") = A.L("3") + A.L("4");
        A.L("4") = 0.0;
    );
);

* ---- CHECK FOR T2 =0 ----
IF (T.L("2") LE .0000002,
    A.L("1")=A.L("1") + A.L("2");
    A.L("2")=A.L("3");
    A.L("3")=A.L("4");
    A.L("4") = 0.0;
    T.L("2") = T.L("3");

```

```

        T.L("3")=T.L("4");
    );
*   --- Check for T2=T3 ---
    IF ((T.L("3") - T.L("2")) LE .0000002,
        A.L("2") = A.L("2") + A.L("3");
        A.L("3") = A.L("4");
        A.L("4") = 0.0;
        T.L("3") = T.L("4");
    );
*   --- CHECK FOR A2=0 ---
    IF (A.L("2") LE .0000002,
        A.L("2") = A.L("3");
        A.L("3") = A.L("4");
        A.L("4") = 0.0;
        T.L("2") = T.L("3");
        T.L("3") = T.L("4");
    );
*   --- CHECK FOR A3=0 ---
    IF (A.L("3") LE .0000002,
        A.L("3") = A.L("4");
        A.L("4") = 0.0;
        T.L("3") = T.L("4");
    );

    PUT F1;
    PUT @1,V:5:3,@7,IN:6:4,@14,Z:4:2,@19,A.L("1"):9:7,@29,
        A.L("2"):9:7,@39,A.L("3"):9:7,@49,A.L("4"):9:7/;
    IF (SI1HUMP.MODELSTAT NE 2,
        PUT 'PUKE'/;
    );
    PUT F2;
    PUT @1,V:5:3,@7,IN:6:4,@14,Z:4:2,@19,T.L("2"):9:7,@29,
        T.L("3"):9:7,@39,T.L("4"):9:7/;
    IF (SI1HUMP.MODELSTAT NE 2,
        PUT 'PUKE'/;
    );
);
SLV = 0.0;
Z=Z+0.01;
);
OPTSOLN = 0.0;
IN=IN+.005;
);
V=V+.01;
);

```

IMP23TR.gms

```

$TITLE imp23tr
$OFFUPPER
* THIS PROGRAM SOLVES FOR THE TRANSITION FROM 2 TO 3 IMPULSES
* IN THE O HUMP REGION
* TURN OFF MOST STUFF PRINTED TO .LST FILE
$OFFSYMREF OFFSYMLIST OFFUELLIST OFFUELXREF
OPTION SOLPRINT = OFF;
OPTION LIMROW = 0;
OPTION LIMCOL = 0;

```

SCALAR

WDH Damped frequency in Hertz /1.0/
 Z Damping ratio /0.0/
 PI Pi /3.1415926/
 SOLV IF WE HAVE TRANS SOLV=0 /1.0/
 V Vibration Limit /0.05/
 INS STARTING INSENSITIVITY /0.06/
 IN INSENSITIVITY(INCREMENTS) /0.06/;

SETS

I counter /1*3/
 N Number of LIMITED frequencies /1*2/
 ICT LOOP THRU INSENSITIVITY /1*150/
 ZCT variable to loop thru Z /1*2/
 ILAST(I) LAST COUNTER;
 ILAST(I) = YES\$(ORD(I) EQ CARD(I));

PARAMETER

WP(N) LIMITED FREQUENCIES IN RADIANES
 WD Damped natural frequency in radians;
 WD = 2.0 * PI * WDH ;

VARIABLES

A(I) amplitude of the i th pulse
 T(I) time of the ith pulse
 S cost -TIME OF LAST IMPULSE;

POSITIVE VARIABLE A ;
 POSITIVE VARIABLE T ;
 T.UP(I) = 1.1;
 A.UP(I) = 1.0;
 OPTION DECIMALS=7

EQUATIONS

COST(I) MINIMIZE T(ILAST)
 ORDER(I) ENFORCE TIME ORDER OF IMPULSES
 INITIAL SET T("1") =0
 VIBLIMIT(N,I) Set vibration limit at the freqs. of interest
 AREA Sum of the pulses must be 1;

ORDER(I+1) .. T(I) =L= T(I+1);
 INITIAL .. T("1") =E= 0.0;
 COST(ILAST) .. S =E= T(ILAST);
 VIBLIMIT(N,ILAST) .. $\text{EXP}(-Z*WP(N)*T(ILAST))*\text{SQRT}(\text{SQR}(\text{SUM}(I,A(I)*\text{EXP}(Z*WP(N)*T(I))*\text{SIN}(WP(N)*\text{SQRT}(1.0 - \text{SQR}(Z))*T(I)))) + \text{SQR}(\text{SUM}(I,A(I)*\text{EXP}(Z*WP(N)*T(I))*\text{COS}(WP(N)*\text{SQRT}(1.0 - \text{SQR}(Z))*T(I)))) =L= V ;$
 AREA .. SUM(I,A(I)) =E= 1.0 ;

MODEL IMP23TR /ALL/ ;

FILE F1 output file /IMP23TR.AMP/;
 FILE F2 output file /IMP23TR.TME/;

LOOP(ZCT,

* ---- RESET GUESSES ----
 A.L("1") = 0.5;
 A.L("2") = 0.5;

```
A.L("3") = 0.0;
T.L("2") = 0.52;
T.L("3") = 1.0;
```

* ---APPROXIMATE TRANSITION FROM 2 TO 3 IMPULSES

```
IN=0.0635+0.081979*Z+0.25254*SQR(Z);
```

```
LOOP(ICT,
```

```
  WP("1") = (WD/SQRT(1.0-SQR(Z)))*(1.0-(0.5*IN));
```

```
  WP("2") = (WD/SQRT(1.0-SQR(Z)))*(1.0+(0.5*IN));
```

```
  IF (SOLV GE 0.5,
```

```
    SOLVE IMP23TR USING NLP MINIMIZING S ;
```

* --- Check for an unnecessary third impulse ---

```
  IF ((T.L("3") - T.L("2")) LE .000001,
```

```
    A.L("2") = A.L("2") + A.L("3");
```

```
    A.L("3") = 0.0;
```

```
    T.L("3") = 0.0;
```

```
  );
```

```
  IF (A.L("2") LE .000001,
```

```
    A.L("2") = A.L("3");
```

```
    A.L("3") = 0.0;
```

```
    T.L("2") = T.L("3");
```

```
    T.L("3") = 0.0;
```

```
  );
```

* --- IF WE FOUND TRANS SET SOLV TO ZERO AND SKIP REST OF IN

```
  IF (IMP23TR.MODELSTAT EQ 2,
```

```
    IF(A.L("3") GE 0.00001,
```

```
      SOLV=0.0;
```

```
    );
```

```
  );
```

```
  PUT F1;
```

```
  PUT @1,V:4:2,@6,IN:8:5,@15,Z:4:2,@20,A.L("1"):9:7,@30,
```

```
    A.L("2"):9:7,@40,A.L("3"):9:7/;
```

```
  IF (IMP23TR.MODELSTAT NE 2,
```

```
    PUT 'PUKE'/;
```

```
  );
```

```
  PUT F2;
```

```
  PUT @1,V:4:2,@6,IN:8:5,@15,Z:4:2,@20,T.L("2"):9:7,@30,
```

```
    T.L("3"):9:7/;
```

```
  IF (IMP23TR.MODELSTAT NE 2,
```

```
    PUT 'PUKE'/;
```

```
  );
```

* --- HELP A LOST PROGRAM GET BACK ON TRACK

```
  A.L("1")=0.5;
```

```
  A.L("2")=0.05;
```

```
  A.L("3")=0.45;
```

```
  T.L("2")=0.28;
```

```
  T.L("3")=0.58;
```

```
  );
```

```
);
```

```
IN=IN+0.00001;
```

```
SOLV=1.0;
```

```
Z=Z+.01;
```

```
);
```

HUMP01TR.gms

\$TITLE HUMP01TR

\$OFFUPPER

* THIS PROGRAM SOLVES FOR THE INSENSITIVITY AT WHICH THE
 * SI SHAPER TRANSITIONS FROM 0 HUMPS TO 1 HUMP AS
 * A FUNCTION OF ZETA.
 * TURN OFF PRINTING OF MAPS AND SOLVE STATEMENT IN .LST FILE

\$OFFSYMREF \$OFFSYMLIST \$OFFUELLIST \$OFFUELXREF

OPTION SOLPRINT = OFF;

OPTION LIMROW = 0;

OPTION LIMCOL = 0;

SCALAR

WDH Damped frequency in Hertz /1.0/
 ZETA Damping ratio /0.0/
 PI Pi / 3.1415926 /
 V Vibration Limit /0.02/;

SETS

I counter / 1*3 /
 N Number of LIMITED frequencies /1*2/
 VCT LOOP THRU V /1*11/
 ZCT variable to loop thru zeta /1*33/
 ILAST(I) LAST COUNTER;
 ILAST(I) = YES\$(ORD(I) EQ CARD(I));

PARAMETER

WD Damped natural frequency in radians;
 WD = 2.0 * PI * WDH ;

VARIABLES

IN INSENSITIVITY
 WP(N) LIMITED FREQUENCIES IN RADIANS
 A(I) amplitude of the i th pulse
 T(I) time of the ith pulse
 S cost -TIME OF LAST IMPULSE;

POSITIVE VARIABLE A ;
 POSITIVE VARIABLE T ;
 POSITIVE VARIABLE IN ;
 POSITIVE VARIABLE WP ;

T.UP(I) = 1.0;
 A.UP(I) = 1.0;
 WP.UP(N) = 3*WD;
 IN.UP = 2;
 IN.LO = 0.02;

OPTION DECIMALS=7

EQUATIONS

COST(I) MINIMIZE T(ILAST)
 ORDER(I) ENFORCE TIME ORDER OF IMPULSES
 INITIAL SET T("1") =0
 VIBLIMIT(N,I) Set vibration limit at the freqs. of interest
 AREA Sum of the pulses must be 1
 ZERODERIV Derivative constraint
 EWP1 SETS WP1 BASED ON INSENSITIVITY
 EWP2 SETS WP2 BASED ON INSENSITIVITY;

ORDER(I+1) .. T(I) =L= T(I+1);
 INITIAL .. T("1") =E= 0.0;
 COST(ILAST) .. S =E= T(ILAST) ;

```

VIBLIMIT(N,ILAST)
.. EXP(-ZETA*WP(N)*T(ILAST))*SQR(SQR(SUM(I,A(I)*EXP(ZETA*WP(N)*T(I))*
SIN(WP(N)*SQR(1.0 - SQR(ZETA))*T(I)))) + SQR(SUM(I,A(I)*EXP(ZETA*WP(N)*T(I))*
COS(WP(N)*SQR(1.0 - SQR(ZETA))*T(I)))) =E= V ;
AREA .. SUM(I,A(I)) =E= 1.0 ;
ZERODERIV(ILAST) ..
-ZETA*T(ILAST)*(SQR(SUM(I,A(I)*EXP(ZETA*WP("2")*T(I))*COS(WP("2")*SQR(1.0 -
SQR(ZETA))*T(I))))
+SQR(SUM(I,A(I)*EXP(ZETA*WP("2")*T(I))*SIN(WP("2")*SQR(1.0 - SQR(ZETA))*T(I))))
+ SUM(I,A(I)*EXP(ZETA*WP("2")*T(I))*SIN(WP("2")*SQR(1.0 - SQR(ZETA))*T(I))*
(SUM(I,A(I)*T(I)*SQR(1-SQR(ZETA))*EXP(ZETA*WP("2")*T(I))*COS(WP("2")*SQR(1.0 -
SQR(ZETA))*T(I))
+ SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP("2")*T(I))*SIN(WP("2")*SQR(1.0 -
SQR(ZETA))*T(I))))
+ SUM(I,A(I)*EXP(ZETA*WP("2")*T(I))*COS(WP("2")*SQR(1.0 - SQR(ZETA))*T(I))
*(SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP("2")*T(I))*COS(WP("2")*SQR(1.0 -
SQR(ZETA))*T(I)) -
SUM(I,A(I)*T(I)*SQR(1-SQR(ZETA))*EXP(ZETA*WP("2")*T(I))*SIN(WP("2")*SQR(1.0 -
SQR(ZETA))*T(I))))
=E= 0.0 ;
EWP1 .. WP("1") =E= (WD*(1.0-(0.5*IN)))/SQR(1.0-SQR(ZETA));
EWP2 .. WP("2") =E= (WD*(1.0+(0.5*IN)))/SQR(1.0-SQR(ZETA));

```

```

MODEL HUMP01TR /ALL/ ;
FILE F1 output file /HUMP01TR.OUT/

```

```

LOOP(VCT,
  AL("1") = (1+V)/4;
  AL("2") = (1-V)/2;
  AL("3") = (1+V)/4;
  TL("2") = 0.45;
  TL("3") = 0.9;
  INL = .17+V;
  WPL("1") =0.89*WD/SQR(1.0-SQR(ZETA));
  WPL("2") =1.11*WD/SQR(1.0-SQR(ZETA));
  PUT F1;
  ZETA=0.0
  LOOP(ZCT,
    SOLVE HUMP01TR USING NLP MINIMIZING S ;
    PUT @1,V:4:2,@6,INL:8:6,@15,ZETA:5:3,@22,AL("1"):8:6,@31,
      AL("2"):8:6,@40,AL("3"):8:6,@49,TL("2"):8:6,
      @58,TL("3"):8:6;
    IF (HUMP01TR.MODELSTAT NE 2,
      PUT 'PUKE'/;
    );
    ZETA=ZETA+.01;
  );
  V=V+.01;
);

```

FS.gms

```
$TITLE fs (NEG,SEQ=1)
```

```
$OFFUPPER
```

```
* This program IMPLEMENTS FREQUENCY SAMPLING.
```

```
* The constraints are for an arbitrarily insensitive shaper
```

```
* TWO PEAKS AND 3 ZERO SLOPE FREQUENCIES
```

```
SCALAR
```

WDH Damped frequency in Hertz /1/
 ZETA Damping ratio /0.0/
 PI Pi / 3.1415926 /
 V Vibration Limit /.05 /;

SETS

I counter / 1*5 /
 N Number of LIMITED frequencies /1*21/
 ILAST(I) LAST COUNTER;
 ILAST(I) = YES\$(ORD(I) EQ CARD(I));

PARAMETER

WL(N) FREQUENCIES TO LIMIT IN PERCENTAGE OF WDH
 /1 0.80
 2 0.93
 3 0.94
 4 0.95
 5 0.96
 6 0.97
 7 0.98
 8 0.985
 9 0.99
 10 0.995
 11 1.00
 12 1.01
 13 1.02
 14 1.03
 15 1.04
 16 1.05
 17 1.06
 18 1.07
 19 1.08
 20 1.09
 21 1.20/

WP(N) LIMITED FREQUENCIES IN RADIAN
 W undamped natural frequency
 WD Damped natural frequency in radians
 DELTAW approx distance to touchdown freq. from W;
 WD = 2.0 * PI * WDH ;
 W = WD / SQRT(1.0 - SQR(ZETA)) ;
 WP(N) = WL(N)*W;

VARIABLES

A(I) amplitude of the i th pulse
 T(I) time of the ith pulse
 S cost -TIME OF LAST IMPULSE
 COSSUM(N) SUM AIEXPCOS
 SINSUM(N) SUM AIEXPSIN
 TZCOSSUM(N) SUM AI*TI*ZETA*EXP*COS
 TZSINSUM(N) SUM AI*TI*ZETA*EXP*SIN
 TSQRTZCOS(N) SUM AI*TI*SQRT1-Z2*EXP*COS
 TSQRTZSIN(N) SUM AI*TI*SQRT1-Z2*EXP*SIN;
 POSITIVE VARIABLE A ;
 POSITIVE VARIABLE T ;
 * INITIAL GUESSES **
 T.L("2") = 0.5/WDH;
 T.L("3") = 1.0/WDH;
 T.L("4") = 1.25/WDH;

```

T.L("5") = 1.54/WDH;
A.L("1") = 0.2625;
A.L("2") = 0.475;
A.L("3") = 0.2625;
A.L("4") = 0.0;
A.L("5") = 0.0;
COSSUML("1") = V-.01;
COSSUML("2") = V-.01;
TSQRTZCOS.L("1") = (.5/WDH)*((1.0-V)/2)*COS(W*.5/WDH) +
(1.0/WDH)*((1.0+V)/4)*COS(W*1.0/WDH);
TSQRTZCOS.L("2") = (.5/WDH)*((1.0-V)/2)*COS(W*.5/WDH) +
(1.0/WDH)*((1.0+V)/4)*COS(W*1.0/WDH);
OPTION DECIMALS=6

```

EQUATIONS

```

COST(I)           maximize WZ("3")-WZ("1")
ORDER(I)         ENFORCE TIME ORDER OF IMPULSES
INITIAL          SET T("1") =0
VIBLIMIT(N,I)   set vibration limit at the freqs. of interest
AREA            sum of the pulses must be constant
ECOSSUM(N)      SUM AIEXPCOS
ESINSUM(N)      SUM AIEXPSIN
ETZCOSSUM(N)    SUM AI*TI*ZETA*EXP*COS
ETZSINSUM(N)    SUM AI*TI*ZETA*EXP*SIN
ETSQRTZCOS(N)   SUM AI*TI*SQRT1-Z2*EXP*COS
ETSQRTZSIN(N)   SUM AI*TI*SQRT1-Z2*EXP*SIN;

```

```

ORDER(I+1) .. T(I) =L= T(I+1);
INITIAL .. T("1") =E= 0.0;
COST(ILAST) .. S =E= T(ILAST) ;
ECOSSUM(N) .. COSSUM(N) =E= SUM(I,A(I)*EXP(ZETA*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 -
SQR(ZETA))*T(I))) ;
ESINSUM(N) .. SINSUM(N) =E= SUM(I,A(I)*EXP(ZETA*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 -
SQR(ZETA))*T(I))) ;
ETZCOSSUM(N) .. TZCOSSUM(N) =E=
SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 - SQR(ZETA))*T(I))) ;
ETZSINSUM(N) .. TZSINSUM(N) =E=
SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 - SQR(ZETA))*T(I))) ;
ETSQRTZCOS(N) .. TSQRTZCOS(N) =E= SUM(I,A(I)*T(I)*SQRT(1-
SQR(ZETA))*EXP(ZETA*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 - SQR(ZETA))*T(I))) ;
ETSQRTZSIN(N) .. TSQRTZSIN(N) =E= SUM(I,A(I)*T(I)*SQRT(1-
SQR(ZETA))*EXP(ZETA*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 - SQR(ZETA))*T(I))) ;
VIBLIMIT(N,ILAST) .. EXP(-ZETA*WP(N)*T(ILAST))*SQRT(SQR(SINSUM(N)) +
SQR(COSSUM(N))) =L= V ;
AREA .. SUM(I,A(I)) =E= 1.0 ;

```

MODEL DAMP /ALL/ ;

SOLVE DAMP USING NLP MINIMIZING S ;

DISPLAY "Arbitrarily Insensitive", T.L, A.L;

LP-2MOD.gms

\$TITLE LPMOD

\$OFFUPPER

* This PROGRAM FINDS A SEQUENCE USING LEAST SQUARES FOR ONE MODE

* copyright 1994 by Convolve, Inc.

```
* TURN OFF PRINTING OF MAPS AND SOLVE STATEMENT IN LST FILE
*$OFFSYMXREF OFFSYMLIST OFFUELLIST OFFUELXREF
*OPTION SOLPRINT = OFF;
*OPTION LIMROW = 0;
*OPTION LIMCOL = 0;
```

OPTIONS DECIMALS=6

SCALAR

```
DT time spacing /0.05/
PI Pi /3.1415926 /
```

SETS

```
I counter /1*25/
N Number of SAMPLED frequencies /1*20/
J NUMBER OF FREQUENCIES /1*1/
ILAST(I) i LAST COUNTER
NLAST(N) N LAST COUNTER;
ILAST(I) = YES$(ORD(I) EQ CARD(I));
NLAST(N) = YES$(ORD(N) EQ CARD(N));
```

PARAMETER

```
T(I) time of the i th pulse
Z(N) Damping ratio of sample mode N
```

```
/1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 .1
11 .1
12 .1
13 .1
14 .1
15 .1
16 .1
17 .1
18 .1
19 .1
20 .1/
```

WSAMP(N) Sampled frequency data

```
/1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
11 0
12 0
13 0
14 0
```

15 0
 16 0
 17 0
 18 0
 19 0
 20 0/

WPH(N) THE FREQUENCY SAMPLES FOR LEAST SQUARES Hertz

/1 0.8
 2 0.85
 3 0.9
 4 0.95
 5 1.0
 6 1.05
 7 1.10
 8 1.15
 9 1.20
 10 2.3
 11 2.35
 12 2.40
 13 2.45
 14 2.5
 15 2.55
 16 2.6
 17 2.65
 18 2.7
 19 2.75
 20 2.8/

WP(N) Sample Frequency in radians;

T(I) = DT * (ORD(I) - 1);

WP(N) = WPH(N)*2*PI;

VARIABLES

A(I) amplitude of the i th pulse

S cost - vibration limit over range of frequencies

V vibration magnitude

POSITIVE VARIABLE A ;

EQUATIONS

COST second derivative cost function

VIBLIMIT(I) LIMIT EI FREQ

AREA sum of the pulses must be constant;

VIBLIMIT(ILAST)

.. SUM(N,SQR(WSAMP(N) - SQR(EXP(-
 2*Z(N)*WP(N)*T(ILAST))*SQR(SUM(I,A(I)*EXP(Z(N)*WP(N)*T(I))*
 SIN(WP(N)*SQR(1.0 - SQR(Z(N))*T(I)))) + SQR(SUM(I,A(I)*EXP(Z(N)*WP(N)*T(I))*
 COS(WP(N)*SQR(1.0 - SQR(Z(N))*T(I)))))) =E= V ;

COST .. S =E= V ;

AREA .. SUM(I,A(I)) =E= 1.0 ;

MODEL LPMOD /ALL/ ;

SOLVE LPMOD USING NLP MINIMIZING S ;

FILE F1 OUTPUT FILE /LPMOD.OUT/;

PUT F1;

loop (I,

IF (A.L(I) NE 0,

PUT @1,T(I):8:6,@11,A.L(I):8:6/;

);
);

Negative Input Shapers

NEGZV.gms

\$TITLE Neg-ZV

\$OFFUPPER

- * This problem finds a sequence that zeros vibration
- * minimizing the end time of the sequence. uses Negative Sequences.
- * The parameter PEAK is the maximum partial sum of the impulses
- * (See "Design and Implementation of Time-Optimal Negative Input Shapers",
- * Singhose, Singer, and Seering, 1994 ASME Winter Annual Meeting)

SCALAR

PI Pi / 3.1415926 /
PEAK Peak Partial Sum /1.0/ ;

SETS

I counter / 1*3 /
J Number of frequencies /1/
ILAST(I) last counter ;
ILAST(I) = YES\$(ORD(I) EQ CARD(I)) ;

ALIAS(I,K);

PARAMETER

WDH(J) Damped frequencies in Hertz
/1 1.0
/
ZETA(J) Damping ratio of mode J
/1 0.0
/
W(J) undamped natural frequency J
WD(J) Damped natural frequency J in radians ;
WD(J) = 2.0 * PI * WDH(J) ;
W(J) = WD(J) / SQRT(1.0 - SQR(ZETA(J))) ;

VARIABLES

A(I) amplitude of the i th pulse
T(I) time of the i th pulse
S cost - Time of last pulse ;

POSITIVE VARIABLE T ;

A.L("1") = PEAK ;
A.L("2") = -2.0 * PEAK;
A.L("3") = PEAK + 1.0;
T.L("1") = 0.0 ;
T.L("2") = (0.152406+.24141*ZETA("1"))/WDH("1") - (PEAK-1.0)*.03;
T.L("3") = (0.277396+.10332*ZETA("1"))/WDH("1") - (PEAK-1.0)*.04;
OPTION DECIMALS=6

EQUATIONS

INIT	set initial time value
COST(I)	minimize value of last T
ORDER(I)	enforce the ordering of the time values
NOVIB1(J)	no vibration function sin part
NOVIB2(J)	no vibration function cos part

AREA sum of the pulses must be constant
 PARTSUM1(K) partial sum of the pulses less than one
 PARTSUM2(K) partial sum of the pulses greater than neg one;

```
INIT ..    T("1") =E= 0.0 ;
COST(ILAST) .. S =E= T(ILAST) ;
ORDER(I+1) .. T(I) =L= T(I+1) ;
NOVIB1(J) .. SUM(LA(I)*EXP(ZETA(J)*W(J)*T(I))*SIN(WD(J)*T(I))) =E= 0.0 ;
NOVIB2(J) .. SUM(LA(I)*EXP(ZETA(J)*W(J)*T(I))*COS(WD(J)*T(I))) =E= 0.0 ;
AREA ..    SUM(I,A(I)) =E= 1.0 ;
PARTSUM1(K) .. SUM(IS(ORD(I) LE ORD(K)),A(I)) =L= PEAK ;
PARTSUM2(K) .. SUM(IS(ORD(I) LE ORD(K)),A(I)) =G= -PEAK ;
```

MODEL NEGZV /ALL/ ;

SOLVE NEGZV USING NLP MINIMIZING S ;

DISPLAY A.L, A.M ,T.L, T.M ;

```
FILE F1 output file /NEGZV.OUT/;
PUT F1;
PUT @1,A.L("1"):8:6,@10,A.L("2"):8:6,@19,A.L("3"):8:6/;
PUT @1,T.L("1"):8:6,@10,T.L("2"):8:6,@19,T.L("3"):8:6/;
```

NEGZVD.gms

\$TITLE NEGZVD (NEG,SEQ=1)

\$OFFUPPER

- * This problem finds a sequence that zeros vibration and first deriv.
- * minimizing the end time of the sequence. uses Negative Sequences.
- * The parameter PEAK is the amount of temporary peak current allowed

SCALAR

```
PI   Pi    / 3.1415926 /
PEAK  Peak current  /4.0/ ;
```

SETS

```
I  counter  / 1*5 /
J  Number of frequencies /1/
ILAST(I) last counter ;
ILAST(I) = YESS(ORD(I) EQ CARD(I)) ;
```

ALIAS(I,K);

PARAMETER

```
WDH(J) Damped frequencies in Hertz
      /1 1.0
      /
ZETA(J) Damping ratio of mode J
      /1 0.0
      /
W(J) undamped natural frequency J
WD(J) Damped natural frequency J in radians ;
WD(J) = 2.0 * PI * WDH(J) ;
W(J) = WD(J) / SQRT(1.0 - SQR(ZETA(J))) ;
```

VARIABLES

```
A(I) amplitude of the i th pulse
T(I) time of the i th pulse
S   cost - Time of last pulse ;
```


POSITIVE VARIABLE T ;

A.L("1") = PEAK ;
 A.L("2") = -2.0 * PEAK;
 A.L("3") = 2.0 * PEAK;
 A.L("4") = -2.0 * PEAK;
 A.L("5") = PEAK + 1.0;
 T.L("1") = 0.0 ;
 T.L("2") = (0.152406+.24141*ZETA("1"))/WDH("1") - (PEAK-1.0)*.03;
 T.L("3") = (0.277396+.10332*ZETA("1"))/WDH("1") - (PEAK-1.0)*.04;
 T.L("4") = (0.631311+.33061*ZETA("1"))/WDH("1") - (PEAK-1.0)*.05;
 T.L("5") = (0.678948+.18899*ZETA("1"))/WDH("1") - (PEAK-1.0)*.05;

OPTION DECIMALS=6
 DISPLAY A.L, T.L ;

EQUATIONS

INIT	set initial time value
COST(I)	minimize value of last T
ORDER(I)	enforce the ordering of the time values
DERIV1(J)	first derivative function sin part
DERIV2(J)	first derivative function cos part
NOVIB1(J)	no vibration function sin part
NOVIB2(J)	no vibration function cos part
AREA	sum of the pulses must be constant
PARTSUM1(K)	partial sum of the pulses less than one
PARTSUM2(K)	partial sum of the pulses greater than neg one;

INIT .. T("1") =E= 0.0 ;
 COST(ILAST) .. S =E= T(ILAST) ;
 ORDER(I+1) .. T(I) =L= T(I+1) ;
 DERIV1(J) .. SUM(I,A(I)*T(I)*EXP(ZETA(J)*W(J)*T(I))*SIN(WD(J)*T(I)))
 =E= 0.0 ;
 DERIV2(J) .. SUM(I,A(I)*T(I)*EXP(ZETA(J)*W(J)*T(I))*COS(WD(J)*T(I)))
 =E= 0.0 ;
 NOVIB1(J) .. SUM(I,A(I)*EXP(ZETA(J)*W(J)*T(I))*SIN(WD(J)*T(I))) =E= 0.0 ;
 NOVIB2(J) .. SUM(I,A(I)*EXP(ZETA(J)*W(J)*T(I))*COS(WD(J)*T(I))) =E= 0.0 ;
 AREA .. SUM(I,A(I)) =E= 1.0 ;
 PARTSUM1(K) .. SUM(IS(ORD(I) LE ORD(K)),A(I)) =L= PEAK ;
 PARTSUM2(K) .. SUM(IS(ORD(I) LE ORD(K)),A(I)) =G= -PEAK ;

MODEL NEGZVD /ALL/ ;

SOLVE NEGZVD USING NLP MINIMIZING S ;

DISPLAY A.L, A.M ,T.L, T.M ;

NEGEI.gms

\$TITLE NEGEI (NEG,SEQ=1)

\$OFFUPPER

- * This program finds a negative Extra-Insensitive shaper for 1 mode.
- * The time of the last impulse is minimized.
- * In the presence of damping, the symmetry of the touchdown
- * frequencies is sacrificed - giving skewed insensitivity.
- * The vibration is set equal to VLIM at W and the slope of
- * sensitivity curve is forced to zero at that point.
- * Further, the curve is forced to zero on either side of W.

SCALAR

WDH Damped frequency in Hertz /1.0/
 ZETA Damping ratio /0.00/
 PI Pi / 3.1415927 /
 VLIM Vibration limit at WDH /0.02/
 P Input Peak /3.0/;

SETS

I counter / 1*5 /
 L Number of touchdown frequencies /1*2/

PARAMETER

W undamped natural frequency in radians
 WD Damped natural frequency in radians
 A(I) impulse amplitudes
 DELTAW approx distance to touchdown freq. from W;
 $A("1") = P;$
 $A("2") = -2*P;$
 $A("3") = 2*P;$
 $A("4") = -2*P;$
 $A("5") = P+1;$
 $WD = 2.0 * PI * WDH ;$
 $W = WD / \text{SQRT}(1.0 - \text{SQR}(ZETA)) ;$

* The following equation is exact for no damping. The (/P(1-ZETA) term is a hack for damping.
 $\text{DELTAW} = (W/PI)*\text{ARCTAN}(2*\text{SQRT}(VLIM)/(1-VLIM))/(P*(1-ZETA));$

VARIABLES

T(I) time of the i th pulse
 TW(L) touchdown frequencies (undamped)
 WP FREQUENCY OF VIBRATION PEAK
 S cost - distance from TW("2") to TW("1") - maximize this
 COSSUM SUM AIEXPCOS
 SINSUM SUM AIEXPSIN
 TZCOSSUM SUM AI*TI*ZETA*EXP*
 TZSINSUM SUM AI*TI*ZETA*EXP*SIN
 TSQRTZCOS SUM AI*TI*SQRT1-Z2*EXP*
 TSQRTZSIN SUM AI*TI*SQRT1-Z2*EXP*SIN;
 POSITIVE VARIABLE T ;
 POSITIVE VARIABLE TW ;
 POSITIVE VARIABLE WP ;

* LIMITS ON THE TOUCHDOWN FREQUENCIES **

TW.UP("1") = W;
 TW.LO("2") = W;

* INITIAL GUESSES **

TW.L("1") = W-DELTAW;
 TW.L("2") = W+DELTAW;
 WP.L = 1.0*W ;
 T.L("1") = 0.0 ;
 $T.L("2") = (.15048+.26755*ZETA)*\text{EXP}((-37191-.34711*ZETA-.075744*\text{SQR}(ZETA))*\text{LOG}(P))/WDH;$
 $T.L("3") = (.28746+.12505*ZETA)*\text{EXP}((-1618-.01035*ZETA-.23769*\text{SQR}(ZETA))*\text{LOG}(P))/WDH;$
 $T.L("4") = (.61213+.20742*ZETA)*\text{EXP}((-28491-.15455*ZETA-.07634*\text{SQR}(ZETA))*\text{LOG}(P))/WDH;$

```

T.L("5") = (.67002+.03+.08883*ZETA)*EXP((- .24135-.10405*ZETA-
.17039*SQR(ZETA))*LOG(P))/WDH;
* THESE HAVE BEEN MANUALLY OVERRIDEN FOR VLIM=0.02, P=3, Z=0
  T.L("5") = .518;
  T.L("4") = .446;
  T.L("3") = .25;
  T.L("2") = .108;
* THESE MAY HAVE TO BE MODIFIED DEPENDING ON P AND ZETA
  COSSUM.L = 0.0*VLIM/0.05;
  SINSUM.L = -.05*VLIM/0.05;
* TSQRTZCOS.L = 0.004;
  TSQRTZSIN.L = -0.01;
  OPTION DECIMALS=6

```

EQUATIONS

INIT	set initial time value
COST	minimize t5
ORDER(I)	enforce the ordering of the time values
VIBLIMIT	set vibration limit at the freq. of interest
DERIV	deriv constraint
WPLW	INSURES WP Equals W
ECOSSUM	SUM AI*W*EXPCOS
ESINSUM	SUM AI*W*EXPSIN
ETZCOSSUM	SUM AI*TI*ZETA*EXP*COS
ETZSINSUM	SUM AI*TI*ZETA*EXP*SIN
ETSQRTZCOS	SUM AI*TI*SQRT1-Z2*EXP*COS
ETSQRTZSIN	SUM AI*TI*SQRT1-Z2*EXP*SIN
TOUCH1(L)	touchdown of the insensitivity curve
TOUCH2(L)	touchdown of the insensitivity curve;

```

INIT ..   T("1") =E= 0.0 ;
COST ..   S =E= T("5");
ORDER(I+1) .. T(I) =L= T(I+1) ;
WPLW ..   WP =E= W;
ECOSSUM .. COSSUM =E= SUM(I,A(I)*EXP(ZETA*WP*T(I))*COS(WP*SQR(1.0 -
SQR(ZETA))*T(I))) ;
ESINSUM .. SINSUM =E= SUM(I,A(I)*EXP(ZETA*WP*T(I))*SIN(WP*SQR(1.0 -
SQR(ZETA))*T(I))) ;
ETZCOSSUM .. TZCOSSUM =E=
SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP*T(I))*COS(WP*SQR(1.0 - SQR(ZETA))*T(I))) ;
ETZSINSUM .. TZSINSUM =E= SUM(I,A(I)*T(I)*ZETA*EXP(ZETA*WP*T(I))*SIN(WP*SQR(1.0
- SQR(ZETA))*T(I))) ;
ETSQRTZCOS .. TSQRTZCOS =E= SUM(I,A(I)*T(I)*SQR(1-
SQR(ZETA))*EXP(ZETA*WP*T(I))*COS(WP*SQR(1.0 - SQR(ZETA))*T(I))) ;
ETSQRTZSIN .. TSQRTZSIN =E= SUM(I,A(I)*T(I)*SQR(1-
SQR(ZETA))*EXP(ZETA*WP*T(I))*SIN(WP*SQR(1.0 - SQR(ZETA))*T(I))) ;
VIBLIMIT .. EXP(-ZETA*WP*T("5"))*SQR(SQR(SINSUM) + SQR(COSSUM)) =E= VLIM ;

DERIV .. (-ZETA*T("5")*(SQR(COSSUM)+SQR(SINSUM)) + SINSUM*(TSQRTZCOS +
TZSINSUM) + COSSUM*(TZCOSSUM - TSQRTZSIN))/
(EXP(ZETA*WP*T("5"))*SQR(SQR(COSSUM)+SQR(SINSUM))) =E= 0.0 ;
TOUCH1(L) .. SUM(I,A(I)*EXP(ZETA*TW(L)*T(I))*
SIN(TW(L)*SQR(1.0 - SQR(ZETA))*T(I))) =E= 0.0 ;
TOUCH2(L) .. SUM(I,A(I)*EXP(ZETA*TW(L)*T(I))*
COS(TW(L)*SQR(1.0 - SQR(ZETA))*T(I))) =E= 0.0 ;

```

MODEL NEGEI /ALL/ ;

SOLVE DAMP USING NEGEI MINIMIZING S;
 DISPLAY T.L

Constant-Magnitude Shapers

CAPZV.gms

\$TITLE CAPZV

\$OFFUPPER

- * This program finds Constant Amplitude Pulses FOR NO VIBRATION.
- * FOR A TWO MASS- ONE SPRING SYSTEM
- * The time of the Pulse sequence is minimized.

SCALAR

WDH Damped frequency in Hertz /1.0/
 ZETA Damping ratio /0.0/
 PI Pi /3.1415927/
 M TOTAL MASS OF SYSTEM /2/;

SETS

I counter /1*5/
 N NUMBER OF LIMITED FREQS /1*1/
 VCT LOOP THRU VLIM /1*1/
 ZCT variable to loop thru zeta /1*1/
 ILAST(I) LAST COUNTER;
 ILAST(I) = YESS(ORD(I) EQ CARD(I));

PARAMETER

W undamped natural frequency in radians
 WD Damped natural frequency in radians
 A(I) impulse amplitudes;
 A("1")=1;
 A("2")=-2;
 A("3")=2;
 A("4")=-2;
 A("5")=1;
 WD = SQRT(2.0);
 W = WD / SQRT(1.0 - SQR(ZETA)) ;

VARIABLES

T(I) time of the i th pulse
 D(I) WIDTH OF ITH PULSE
 WP(N) FREQUENCY OF VIBRATION PEAK
 S TIME OF T(ILAST);
 POSITIVE VARIABLE T ;
 POSITIVE VARIABLE D ;
 POSITIVE VARIABLE WP ;
 WP.FX("1") = WD;
 OPTION DECIMALS=6;

T.L("2")=1;
 T.L("3")=2;
 T.L("4")=3;
 T.L("5")=4;

DISPLAY T.L;

EQUATIONS

```

INIT          set initial time value
COST(I)       minimize T(ILAST)
ORDER(I)      enforce the ordering of the time values
FINALPOS      MAKES FINAL POSITION =1
T5E2T3       T5=2T3
T4E2T3MT2    T4=2T3-T2
VIBLIMIT(N,I) Set vibration limit at the freqs. of interesT;
    
```

```

INIT ..      T("1") =E= 0.0 ;
COST(ILAST) .. S =E= T(ILAST);
ORDER(I+1) .. T(I) =L= T(I+1);
FINALPOS .. 0.0 =E= 2.0+2*T("2")**2+T("3")**2-4*T("2")*T("3");
T5E2T3 ..   T("5") =E= 2*T("3");
T4E2T3MT2 .. T("4") =E= 2*T("3")-T("2");
VIBLIMIT(N,ILAST)$((ORD(N) EQ 1))
    .. EXP(-ZETA*WP(N)*T(ILAST))*SQRT(SQR(SUM(I,A(I)*EXP(ZETA*WP(N)*T(I))*
    SIN(WP(N)*SQRT(1.0 - SQR(ZETA))*T(I)))) + SQR(SUM(I,A(I)*EXP(ZETA*WP(N)*T(I))*
    COS(WP(N)*SQRT(1.0 - SQR(ZETA))*T(I)))) =E= 0.0 ;
    
```

MODEL CAPZV /ALL/ ;

FILE F1 output file /CAPZV.OUT/;
 PUT F1;

SOLVE CAPZV USING NLP MINIMIZING S ;

```

PUT @1,T.L("2"):8:6,@10,T.L("3"):8:6,@19,T.L("4"):8:6,@28,T.L("5"):8:6/;
IF (CAPZV.MODELSTAT NE 2,
    PUT 'PUKE'/;
);
    
```

CVCAPZV.gms

\$TITLE CVCAPZV

\$OFFUPPER

```

* This program finds Constant Amplitude Pulses FOR NO VIBRATION
* ACCELERATION TO A CONSTANT VELOCITY
* FOR A TWO MASS- ONE SPRING SYSTEM
* The time of the Pulse sequence is minimized.
* TURN OFF PRINTING OF MAPS AND SOLVE STATEMENT IN LST FILE
$OFFSYMXREF $OFFSYMLIST $OFFUELLIST $OFFUELXREF
OPTION SOLPRINT = OFF;
OPTION LIMROW = 0;
OPTION LIMCOL = 0;
    
```

SCALAR

```

WD Frequency in RADIANS      /1.4142136/
Z Damping ratio              /0.0/
DESVEL                      /1.0/
M1 MASS#1                   /1.0/
M2 MASS#2                   /1.0/
PI Pi                        /3.1415927/;
    
```

SETS

```

I counter                    /1*4/
N NUMBER OF LIMITED FREQS    /1*1/
ILAST(I) LAST COUNTER:
ILAST(I) = YES$(ORD(I) EQ CARD(I));
    
```

PARAMETER

W undamped natural frequency in radians
 WD Damped natural frequency in radians
 A(I) impulse amplitudes;
 A("1")=1.0;
 A("2")=-1.0;
 A("3")=1.0;
 A("4")=-1.0;

VARIABLES

T(I) time of the i th pulse
 WP(N) FREQUENCY OF VIBRATION PEAK
 S TIME OF T(ILAST);
 POSITIVE VARIABLE T ;
 POSITIVE VARIABLE WP ;
 WP.FX("1") = WD;

OPTION DECIMALS=6;

T.L("2")=1.437;
 T.L("3")=2.516;
 T.L("4")=3.597;

DISPLAY T.L;

EQUATIONS

INIT set initial time value
 COST(I) minimize T(ILAST)
 ORDER(I) enforce the ordering of the time values
 DESIREDVEL MAKES DESIRED VELOCITY = DESVEL
 ZEROSIN(N) ZERO THE SIN COMPONENT OF VIBRATION
 ZEROCOS(N) ZERO THE COS COMPONENT OF VIBRATION;

INIT .. T("1") =E= 0.0 ;
 COST(ILAST) .. S =E= T(ILAST);
 ORDER(I+1) .. T(I) =L= T(I+1) ;
 DESIREDVEL .. DESVEL =E= (T("2")-T("3")+T("4"))/(M1+M2);
 ZEROSIN(N) ..
 SUM(I,A(I)*EXP(Z*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I))) =E= 0.0;
 ZEROCOS(N) ..
 SUM(I,A(I)*EXP(Z*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 - SQR(Z))*T(I))) =E= 0.0 ;

MODEL ZV /ALL/ ;

FILE F1 output file /CVCAPZV.OUT/;
 PUT F1;

SOLVE ZV USING NLP MINIMIZING S ;

PUT @1,DESVEL:5:3,@8,T.L("2"):8:6,@17,T.L("3"):8:6,@26,T.L("4"):8:6/;
 IF (ZV.MODELSTAT NE 2,
 PUT 'PUKE'/;
);

CAPEI.gms

\$TITLE CAPEI
 \$OFFUPPER

* This program finds Constant Amplitude Pulses FOR NO VIBRATION

- * and EXTRA-INSENSITIVE CONSTRAINTS.
- * FOR A TWO MASS- ONE SPRING SYSTEM
- * The time of the Pulse sequence is minimized.

SCALAR

WDH Frequency in HERTZ /2.0/
 Z Damping ratio /0.0/
 V VIBRATION LIMIT /0.05/
 FINPOS FINAL POSITION /,07032/
 PI Pi /3.1415927/;

SETS

I counter /1*7/
 N NUMBER OF LIMITED FREQS /1*3/
 ILAST(I) LAST COUNTER;
 ILAST(I) = YESS(ORD(I) EQ CARD(I));

PARAMETER

W undamped natural frequency in radians
 WD Damped natural frequency in radians
 A(I) impulse amplitudes;
 A("1")=1.0;
 A("2")=-2.0;
 A("3")=2.0;
 A("4")=-2.0;
 A("5")=2.0;
 A("6")=-2.0;
 A("7")=1.0;
 WD=WDH*2*PI;

VARIABLES

T(I) time of the i th pulse
 WP(N) FREQUENCY OF VIBRATION PEAK
 S TIME OF T(ILAST);
 POSITIVE VARIABLE T ;
 POSITIVE VARIABLE WP ;
 WPL("1") = .95*WD;
 WPL("2") = WD;
 WPL("3") = 1.05*WD;
 WPL("4") = 0.7*WD;
 WPL("5") = 1.3*WD;
 OPTION DECIMALS=6;

T.L("2")=0.179;
 T.L("3")=0.236;
 T.L("4")=0.441;
 T.L("5")=0.646;
 T.L("6")=0.703;
 T.L("7")=0.882;
 T.UP(I) = 1.2;

EQUATIONS

INIT	set initial time value
COST(I)	minimize T(ILAST)
ORDER(I)	enforce the ordering of the time values
ORDER WP(N)	ORDER WPS
FINALPOS	MAKES FINAL POSITION = DESIRED
ZEROSUM	SUM OF PULSES EQUAL ZERO

VIBLIMIT(N,I) Set vibration limit at the freqs. of interesT
 ZVIBLIMIT(N,I) ZERO vibration at the freqs. of interesT
 ZERODERIV(N,I) ZERO THE DERIVATIVE AT WD;

```

INIT .. T("1") =E= 0.0 ;
COST(ILAST) .. S =E= T(ILAST);
ORDERWP(N+1) .. WP(N) =L= WP(N+1);
ORDER(I+1) .. T(I) =L= T(I+1) ;
FINALPOS .. FINPOS =E= 0.5*(-T("2")**2) + T("3")**2 - (T("4")**2
    +T("5")**2 -T("6")**2) -0.5*T("7")**2
    +2*T("2")*T("7") -2*T("3")*T("7")+2*T("4")*T("7")
    -2*T("5")*T("7") + 2*T("6")*T("7"));
ZEROSUM .. 0.0 =E= T("2")-T("3")+T("4")-T("5")+T("6")-0.5*T("7");
VIBLIMIT(N,ILAST)$(ORD(N) EQ 2)
    .. EXP(-Z*WP(N)*T(ILAST))*SQRT(SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*
    SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I)))) + SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*
    COS(WP(N)*SQRT(1.0 - SQR(Z))*T(I)))))) =E= V ;
ZVIBLIMIT(N,ILAST)$(ORD(N) EQ 1) OR (ORD(N) EQ 3))
    .. EXP(-Z*WP(N)*T(ILAST))*SQRT(SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*
    SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I)))) + SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*
    COS(WP(N)*SQRT(1.0 - SQR(Z))*T(I)))))) =L= 0.0001 ;
ZERODERIV(N,ILAST)$(ORD(N) EQ 2)
    .. -EXP(-T(ILAST)*WP(N)*Z)*{
    Z*T(ILAST)*SQRT(SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 -
    SQR(Z))*T(I))))
    + SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I))))
    + { SUM(I,A(I)*EXP(Z*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 - SQR(Z))*T(I))
    *SUM(I,A(I)*T(I)*EXP(T(I)*WP(N)*Z)*(SQRT(1-SQR(Z))*SIN(T(I)*WP(N)*SQRT(1-SQR(Z)))
    -Z*COS(T(I)*WP(N)*SQRT(1-SQR(Z))))
    - SUM(I,A(I)*EXP(Z*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 - SQR(Z))*T(I))
    *SUM(I,A(I)*T(I)*EXP(T(I)*WP(N)*Z)*(SQRT(1-SQR(Z))*COS(T(I)*WP(N)*SQRT(1-SQR(Z)))
    +Z*SIN(T(I)*WP(N)*SQRT(1-SQR(Z)))) )
    /(SQRT(SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*COS(WP(N)*SQRT(1.0 -SQR(Z))*T(I))))
    + SQR(SUM(I,A(I)*EXP(Z*WP(N)*T(I))*SIN(WP(N)*SQRT(1.0 -SQR(Z))*T(I)))))) }
    =E= 0.0;

MODEL CAPEI /ALL/ ;

FILE F1 output file /CAPEI.OUT/;
PUT F1;

WD=WDH*2*PI;
WPFX("2") = WD;

SOLVE CAPEI USING NLP MINIMIZING S ;

PUT @1,WDH:6:4,@10,T.L("2"):8:6,@19,T.L("3"):8:6,@28,
    T.L("4"):8:6,@37,T.L("5"):8:6,@46,
    T.L("6"):8:6,@55,T.L("7"):8:6/;

PUT @1,WDH:6:4,@10,T.L("2"):8:6,@19,(T.L("3")-T.L("2")):8:6,@28,
    (T.L("4")-T.L("3")):8:6,@37,(T.L("5")-T.L("4")):8:6,@46,
    (T.L("6")-T.L("5")):8:6,@55,(T.L("7")-T.L("6")):8:6/;
IF (CAPEI.MODELSTAT NE 2,
    PUT 'PUKE'/;
);
    
```


MATLAB Input Shaping™ Toolbox

Copyright Convolve, Inc., 1995

This document describes Convolve's Input Shaping Toolbox. The functions described are used for generating vibration-reducing commands to dynamic systems. This document contains three sections. First, Every function in the toolbox is listed. Second, a brief description is given for most of the commands. Finally, the header files from each function are listed. The last two sections are available on-line in MATLAB by typing "help Contents" and "help READ_ME".

 Universal_IS_Toolbox

Name	Size	Kind	Label	Last Modified
☐ ACC2TRAP.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:19 PM
☐ ACTTF.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 11:24 PM
☐ CIRCLE.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 11:25 PM
☐ CIRCLEP.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 11:26 PM
☐ CON2DIG.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:19 PM
☐ CONTENTS.M	6K	MATLAB document	—	Mon, Jan 23, 1995, 12:28 AM
☐ CONVOLVE.M	2K	MATLAB document	—	Thu, Aug 19, 1993, 2:29 PM
☐ CRMETRIC.M	4K	MATLAB document	—	Fri, Sep 17, 1993, 11:29 PM
☐ DIG2CON.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:20 PM
☐ DIGEI5.m	4K	MATLAB document	—	Sun, Jan 22, 1995, 11:32 PM
☐ DIGSEQ.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:22 PM
☐ DigZVD5.m	2K	MATLAB document	—	Mon, Jan 23, 1995, 12:00 AM
☐ DVD.M	2K	MATLAB document	—	Sun, Dec 5, 1993, 3:54 PM
☐ EI.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:25 PM
☐ EI2HUMP.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:27 PM
☐ EI2M_DMP.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 9:58 PM
☐ EI2M_UND.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:23 PM
☐ EI3HUMP.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:26 PM
☐ EI_UNDMP.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:25 PM
☐ GETACCEL.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:23 PM
☐ GPLANE.M	2K	MATLAB document	—	Mon, Jan 23, 1995, 12:03 AM
☐ MDSIZERA.M	2K	MATLAB document	—	Sat, Sep 18, 1993, 5:41 PM
☐ MDSIZERJ.M	4K	MATLAB document	—	Mon, Sep 13, 1993, 9:47 PM
☐ NEG2MZVD.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:18 PM
☐ NEGEI.M	4K	MATLAB document	—	Sun, Jan 22, 1995, 8:00 PM
☐ NEGZV.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:05 PM
☐ NEGZVD.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:05 PM
☐ NMASSSYS.M	4K	MATLAB document	—	Sun, Jan 22, 1995, 8:08 PM
☐ NORMP.M	2K	MATLAB document	—	Tue, Jul 13, 1993, 1:43 PM
☐ PRESR.M	4K	MATLAB document	—	Mon, Jan 23, 1995, 12:06 AM
☐ PRETF.M	2K	MATLAB document	—	Fri, Jan 4, 1980, 8:10 AM
☐ RAP2MZV.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:10 PM
☐ RAP2MZVD.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:03 PM
☐ Rapp9Seq.m	4K	MATLAB document	—	Thu, Oct 14, 1993, 10:06 AM
☐ READ_ME.m	22K	MATLAB document	—	Mon, Jan 23, 1995, 12:25 AM
☐ RESID.M	4K	MATLAB document	—	Mon, Jan 23, 1995, 12:08 AM

Universal_IS_Toolbox2

Name	Size	Kind	Label	Last Modified
☐ SENSPLIT.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:13 PM
☐ SENSPLT2.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:15 PM
☐ SEQCONV.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:26 PM
☐ SEQSORT.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:26 PM
☐ SHAPGEN.M	6K	MATLAB document	—	Wed, Nov 3, 1993, 1:15 PM
☐ SHAPRES.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:27 PM
☐ SPEEDCRM.M	2K	MATLAB document	—	Mon, Sep 20, 1993, 3:21 PM
☐ TRAPPROF.M	2K	MATLAB document	—	Sun, Sep 12, 1993, 5:32 PM
☐ TRAP_RUL.M	2K	MATLAB document	—	Mon, Jan 23, 1995, 12:22 AM
☐ TWOMASS.M	2K	MATLAB document	—	Thu, Aug 26, 1993, 6:14 PM
☐ VD.M	2K	MATLAB document	—	Sun, Dec 5, 1993, 3:56 PM
☐ W_DIGSEQ.M	2K	MATLAB document	—	Thu, Sep 9, 1993, 10:20 PM
☐ ZV.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:06 PM
☐ ZVD-2M.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:21 PM
☐ ZVD.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:19 PM
☐ ZVD2MD.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:11 PM
☐ ZVDD.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:20 PM
☐ ZVDDD.M	2K	MATLAB document	—	Sun, Jan 22, 1995, 8:21 PM

Contents.M

```
% Input Shaping Toolbox, Copyright Convolve, Inc 1993 9/9/93
%
%
%   Input Shaping Toolbox
%
% One-mode Shapers
%
% EI: A three-impulse sequence that limits the vibration to V at the
% modeling frequency, (instead of forcing it to zero), to achieve
% larger amounts of insensitivity than the ZVD shaper. Shaper length
% equals 1 period. The shaper works well for  $0 < V < .2$  and  $0 < \zeta < .25$ .
% EI2HUMP: A 4-impulse shaper that has two humps in its sensitivity
% curve. It has the same length as a ZVDD, but has a larger
% insensitivity to modelling errors.
% EI3HUMP: A 5-impulse shaper that has three humps in its sensitivity
% curve. It has the same length as a ZVDDD, but has a larger
% insensitivity to modelling errors.
% EI_UNDMP: An exact EI shaper for 1 UNDAMPED mode. Shaper length
% equals 1 period of vibration.
% DIGEI5: An exact digital EI shaper, using matrix inversion technique.
% DIGZVD5: An digital ZVD shaper calculated by linearizing the
% times to the nearest two time discrete time steps and
% then inverting the matrix of the original equations.
% NEGEI: A negative EI shaper for 1 mode. This shaper is shorter
% than the positive EI. The amount of overcurrenting is
% controlled with the parameter P. For P=1, the negative
% shaper is approximately 30% shorter than the positive EI.
% NEGZV: A negative ZV shaper for 1 mode.
% NEGZVD: A negative ZVD shaper.
% ZV : A two-impulse sequence that gives no vibration at
% the modeling frequency, but has poor insensitivity.
% Shaper length equals 1/2 period of vibration.
% ZVD : A three-impulse sequence that has both zero vibration
% and zero slope of the sensitivity curve at the modeling
% frequency. Shaper length equals 1 period of vibration.
% ZVDD : A four-impulse sequence that has zero vibration and zero 1st
% and 2nd derivatives of the sensitivity curve at the modeling
% frequency. Shaper length equals 1.5 periods of vibration.
% ZVDDD: A five-impulse sequence that has zero vibration and zero 1st
% 2nd, & 3rd derivatives of the sensitivity curve at the modeling
% frequency. Shaper length equals 2 periods of vibration.
%
% Two-mode Shapers
% EI2M_DMP: An EI shaper for 2 modes. The solution comes from a surface
% fit to data calculated with GAMS. The shaper works well over
% the ranges of  $0 < V < .2$  and  $0 < \zeta < .25$ . The shapers for each mode
% are convolved together to give the final shaper.
% EI2M_UND: An exact EI shaper 2 UNDAMPED modes. The shapers for each
% mode are convolved together to give the final shaper.
% NEG2MZVD: A negative 2 mode ZVD shaper based on Rappole eqns.
% This shaper is shorter than a 2 mode shaper formed
```

% by convolving two single-mode shapers together.
% RAP2MZV: A 2 mode ZV shaper based on Rappole equations for 2
% undamped modes. This shaper is shorter than two
% single-mode ZV shaper convolved together. Do not use
% this for frequency ratios over 3.76679 - it will
% result in gains larger than one (overcurrenting).
% RAP2MZVD: A 2 mode ZVD shaper based on Rappole equations for 2
% undamped modes. This shaper is shorter than two
% single-mode ZVD shaper convolved together. Do not use
% this for frequency ratios over 3.76679 - it will
% result in gains larger than one (overcurrenting).
% RAPP9SEQ: An exact digital sequence found with Rappole 2-modes,
% 5-impulse constraints. Uses matrix inversion.
% ZVD-2M : A two-mode shaper that uses two ZVD shapers convolved
% together. Shaper length equals the sum of the two periods.
% For a shorter 2-mode shaper see ZVD2MD.
% ZVD2MD: A two-mode shaper that uses ZVD constraints. The solution
% is based on the result of a curve fit to data from GAMS.
% The shaper length is shorter than the sum of the two periods.
% The shaper works well for $1.2 < \tau < 2.8$ and $0 < z < .15$.

%
%

% Trapezoidal Profiling Utilities

%

% Acc2Trap : Calculate a shaped trapezoidal trajectory (both
% velocity and position) using an acceleration table
% TrapProf : Generate position and velocity commands for a trapezoidal
% velocity profile.

%

%

% Additional Input shaping utilities

%

% Convolve : Covolves Input and Shaper and returns the
% result ShapedInput.
% Dig2con : Put a digital sequence in continuous
% DigSeq : Map a sequence onto digital timing loop
% SeqConv : Convolve two continuous sequences together.
% SeqSort : Sort a continuous sequence into correct
% order and combine impulses at the same time.
% ShapRes : Calculate the residual for a given freq and zeta
% Con2Dig : Puts a continuous sequence in digital sequence form.
%
% SensPlot : Plot the residual over range of frequencies
%
% Dig3Imp : Finds a digital sequence for a ThreeImpSeq by linearizing
% the times to the nearest two time discrete time steps and
% then inverting the matrix of the original equations
% W_DigSeq : Uses Watanabe's formula to split each continuous
% impulse into two digital impulses

```

%
%READ_ME.m
% This file contains the headers from all of the
% functions in the Input Shaping Toolbox.
% For a brief description of all the functions in this
% toolbox, see CONTENTS.M
%
% [T,V,P] = acc2trap(Accel,dt)
%
% Calculate a shaped trapezoidal trajectory (both
% velocity and position) using an acceleration table
% The table is generated by GETACCEL.
%
% Parameters:
% Accel - Acceleration Table from GETACCEL
% dt - time step for system (sec/step)
%
% Returns:
% T - A time vector [0 dt 2dt ....];
% V - A velocity vector.
% P - A position vector.
% All outputs have the same length.
%
% See also GETACCEL.
%
%ACTTF.M
% Actuator laplace transform analysis for systems with and without
% prefilters.
% For a system without a prefilter the transfer function from the
% reference input to the actuator is given by CLS(s)/OLP(s) where
% CLS(s) is the closed loop system response transfer function.
% OLP(s) is the open loop system response transfer function.
% For a system with a prefilter the transfer function from the
% reference input to the actuator is given by PRE(s)*CLS(s)/OLP(s)
% where PRE(s) is the prefilter transfer function.
%
% CIRCLE.M
% plots the root locations of the polynomial around in the unit circle
%
% CIRCLEP.M
% plots the root locations of the polynomial around in the unit circle
%
% con2dig CONTINUOUS2DIGITAL Put a continuous sequence in digital
% sequence form.
%
% dseq = con2dig(cseq,dt)
%
% A continuous sequence is an n*2 matrix with times (sec) in
% the first column and amplitudes in the second column.
% A digital sequence as a vector of amplitudes, where each
% value corresponds to a time step. The vector has length
% n = Time_Of_Last_Impulse/dt.
%

```

```

% Parameters:
% cseq, the continuous sequence. Assumes that the
% continuous sequence is spaced at intervals of dt.
% dt, the time step for the digital sequence.
%
% Returns:
% dseq, the digital sequence.
%
% See also DigitizeSeq, Digital2Continuous
%
% Convolve(Input,Shaper,deltaT) -- Bill Singhose
% Covolves Input and Shaper and returns the result ShapedInput.
% A time vector, T, which has the same number of rows as
% ShapedInput is also returned. T starts at zero and is incremented
% deltaT each step.
% Input can be an nxm matrix, where m is the number of inputs.
% Shaper must be a row or column vector.
%
% Function CRMETRIC(a,b,c,d,T)
% [FIRy,FIRact,PRE,TFIR,resid,s]=crmetric(a,b,c,d,T)
% "Controller Redesign Metric"
% Given a continuous system description [a,b,c,d], where
% the first and second outputs are position and actuator
% effort, computes the FIR components of the response for
% the two outputs and the overall prefilter time period TFIR
%
% DIG2CON Put a digital sequence in continuous sequence form.
%
% cseq = dig2con(dseq,dt)
%
% A continuous sequence is an n*2 matrix with times (sec) in
% the first column and amplitudes in the second column.
% A digital sequence as a vector of amplitudes, where each
% value corresponds to a time step. The vector has length
% n = Time_Of_Last_Impulse/dt.
%
% Parameters:
% dseq, the digital sequence.
% dt, the time step for the digital sequence.
%
% Returns:
% cseq, the continuous sequence.
%
% seq = DIGEI5(dt,freq,zeta,vlim)
% Find an exact digital Extra-Insensitive shaper.
% Uses matrix inversion technique.
%
% digseq - Whit Rappole
% DIGITIZESEQ Map a sequence onto digital timing loop
% dseq = digseq(seq,step)
%
% Uses a linear extrapolation to split each continuous
% impulse into two digital impulses
%

```

```

% seq = DigSeq(dt,freq,zeta)
%
% Finds a digital ZVD shaper by linearizing
% the times to the nearest two time discrete time steps and
% then inverting the matrix of the original equations
%
% seq = DigZVD5(dt,freq,zeta)
% Finds a digital ZVD shaper by linearizing
% the times to the nearest two time discrete time steps and
% then inverting the matrix of the original equations
%
%function dvd(amp,freq,zeta,dt)
% or vd(digital_shaper,freq,zeta,dt)
% creates a vector diagram based on the
% provided f (natural frequency in hertz)
% and the time step (dt) and amplitudes of the pulses
%
% Mapping is -> theta = w*time
% So each vector amp[i],time[i] has a corresponding
% position in the vector diagram.
%
% Currently only handles positive vectors amp = [0 to 1]
%
%% EI(fd,z,V,deltaT) -- Bill Singhose
% Generates an EI shaper for 1 mode.
% The solution comes from a surface fit to data calculated with GAMS.
% The shaper works well over the ranges of 0<V<.2 ans 0<z<.25
% fd - damped frequency (Hz) of vibration being controlled.
% z - damping ratio of vibration being controlled.
% V - vibration limit, for 5% vibration, Vlim=0.05.
% deltaT - time spacing at which input to system is updated.
%
%% EI2HUMP
% Generates a two-hump EI shaper
% f - frequency (Hz) of vibration being controlled.
% V - vibration limit
% deltaT - time spacing at which input to system is updated.
%
% This function generates the exact sequence and then uses
% DigSeq to convert the exact sequence to digital format.
%
%% EI2M_DMP(fd1,fd2,z1,z2,V1,V2,deltaT) -- Bill Singhose
% Generates an EI shaper for 2 modes.
% The solution comes from a surface fit to data calculated with GAMS.
% The shaper works well over the ranges of 0<V<.2 ans 0<z<.25
% The shapers for each mode are convolved together to give the
% final shaper.
% fd1,fd2 - frequency (Hz) of damped vibration being controlled.
% zeta1,zeta2 - damping ratios of vibration being controlled.
% Vlim1,Vlim2 - vibration limits, for 5% vibration, Vlim=0.05.
% deltaT - time spacing at which input to system is updated.
%
%% EI2M_UND(f1,f2,Vlim1,Vlim2,deltaT) -- Bill Singhose
% Generates an EI shaper 2 UNDAMPED modes.

```



```

% (The vibration at the modeling frequency is limited to Vlim
% and the insensitivity is maximized)
% The shapers for each mode are convolved together to give the
% final shaper.
% f1,f2 - frequency (Hz) of vibration being controlled.
% Vlim1,Vlim2 - vibration limits, for 5% vibration, Vlim=0.05.
% deltaT - time spacing at which input to system is updated.
%
%% EI3HUMP
% Generates a three-hump EI shaper
% f - frequency (Hz) of vibration being controlled.
% V - vibration limit
% deltaT - time spacing at which input to system is updated.
%
% This function generates the exact sequence and then uses
% DigSeq to convert the exact sequence to digital format.
%
%% EI_UNDMP(f,Vlim,deltaT) -- Bill Singhose
% Generates an EI shaper for 1 UNDAMPED mode.
% (The vibration at the modeling frequency is limited to Vlim
% and the insensitivity is maximized)
% f - frequency (Hz) of vibration being controlled.
% Vlim - vibration limit, for 5% vibration, Vlim=0.05.
% deltaT - time spacing at which input to system is updated.
%
%% getaccel(seq,vmax,amax,des,dt)
%
% Get an acceleration matrix for calculating shaped
% trapezoidal trajectories using changing accelerations.
%
%% GPLANE.M
% G plane analysis of the response of a second order system to damped
% sinusoid inputs.
%
%% MDSIZER - Point to Point move profile based on a given Acceleration Pulse
% [x,v,a,nd]=mdsizer(Dist,AccSeq,T)
% For an AccSeq [1 1 1 ... 1 1 1], mdsizera generates a trapezoidal
% velocity profile.
%
% INPUTS:
%
% Dist - move distance
% AccSeq - acceleration sequence
% T - sample period
%
% OUTPUTS:
%
% x - position
% v - velocity
% a - acceleration
% nd - sample delay before start of second pulse.
%
% NOTE: the derivation involved using the final value theorem and in
% the model used, zero order holds were assumed.

```

```

%
% An arbitrary acceleration pulse is combined with a negative and delayed
% (by nd samples) version of the pulse to generate the overall
% acceleration profile. If the desired move distance (dist) cannot be
% achieved with an integer delay (nd), then the delay is rounded up to
% the next integer value and the acceleration pulse scaled down to meet
% the desired move distance (dist). Note that for very short moves, (nd)
% approaches zero, and the length of the overall acceleration profile
% approaches that of the arbitrary acceleration pulse. This function is
% designed to be used with acceleration pulses that have built in
% vibration cancellation and this is why the pulse shape is not modified
% in the case of short moves.
% Note, the length of the sequences [x], [v], and [a] are different due
% to a pole/zero cancellation that occurs during integration.
%
%% MDSIZERJ - Point to Point move profile based on a given Jerk Sequence
%   [x,v,a,nd]=mdsizer(Dist,JerkSeq,T)
%   For a JerkSeq [1 0 0 ... 0 0 -1], mdsizer generates a trapezoid
%   velocity profile.
%
% INPUTS:
%
%   Dist - move distance
%   JerkSeq - jerk sequence
%   T - sample period
%
% OUTPUTS:
%
%   x - position
%   v - velocity
%   a - acceleration
%   nd - number of sample delays between start of pulses
%
% NOTE: the derivation involved using the final value theorem and in
% the model used, zero order holds were assumed.
%
% An arbitrary acceleration pulse is combined with a negative and delayed
% (by nd samples) version of the pulse to generate the overall
% acceleration profile. If the desired move distance (dist) cannot be
% achieved with an integer delay (nd), then the delay is rounded up to
% the next integer value and the acceleration pulse scaled down to meet
% the desired move distance (dist). Note that for very short moves, (nd)
% approaches zero, and the length of the overall acceleration profile
% approaches that of the arbitrary acceleration pulse. This function is
% designed to be used with acceleration pulses that have built in
% vibration cancellation and this is why the pulse shape is not modified
% in the case of short moves.
% Note, the length of the sequences [x], [v], and [a] are different due
% to a pole/zero cancellation that occurs during integration.
%
%% NEG2MZVD(f1,dt)
% NEG2MZVD Generates a negative 2 mode ZVD shaper
% based on Rappole equations
%

```

```

% f1 - Frequency to shape for (Hz.)
% deltaT - Time spacing (sec/step)
%
%% NEGEI(f,zeta,V,P,deltaT)-- Bill Singhose
% Generates a negative Extra-Insensitive shaper for 1 mode.
%
% Returns the digital sequence by default, and
% the exact sequence as the second variable:
% eg: to generate the exact shaper, use:
%       [dig_shaper,exact_shaper]=singer13(f,z,0.01)
%       note: the 0.01 is ignored for the exact shaper and
%             dig_shaper can be ignored
%
% to Return the digital sequence, use:
%       dig_shaper=singer13(f,z,0.001)
%
% f - frequency (Hz) of vibration being controlled.
% zeta - damping ratio of vibration being controlled.
% V - Vibration Limit
% P - Input Peak
% deltaT - time spacing at which input to system is updated.
%
% This function generates the exact sequence and then uses
% DigitizeSeq to convert the exact sequence to digital format.
%
%% negzv(f,zeta,deltaT)-- Bill Singhose
% Generates a negative ZV shaper for 1 mode.
%
% Returns the digital sequence by default, and
% the exact sequence as the second variable:
%
% to Return the digital sequence, use:
%       dig_shaper=negzv(f,z,0.001)
%
% f - frequency (Hz) of vibration being controlled.
% zeta - damping ratio of vibration being controlled.
% P - Input Peak
% deltaT - time spacing at which input to system is updated.
%
% This function generates the exact sequence and then uses
% DigitizeSeq to convert the exact sequence to digital format.
%
%% NEGZVD(f,zeta,deltaT)-- Bill Singhose
% Generates a negative ZVD shaper for 1 mode.
%
% Returns the digital sequence by default, and
% the exact sequence as the second variable:
% eg: to generate the exact shaper, use:
%       [dig_shaper,exact_shaper]=singer13(f,z,0.01)
%       note: the 0.01 is ignored for the exact shaper and
%             dig_shaper can be ignored
%
% to Return the digital sequence, use:
%       dig_shaper=singer13(f,z,0.001)

```

```

%
% f - frequency (Hz) of vibration being controlled.
% zeta - damping ratio of vibration being controlled.
% P - Input Peak
% deltaT - time spacing at which input to system is updated.
%
%% NMASSSYS [a,b,c,d]=nmasssys(m,k,gains)
% Creates a state space model of an n mass system and returns
% the [a,b,c,d] matrices of the system.
% [m] is a vector of masses (the input force acts on mass m(1)).
% [gains] is a vector of feedback gains, (+) for neg feedback;
% to build a model without feedback use a null matrix []
% for gains.
%
% [k] is a vector of springs between the masses; k(1) is
% between masses m(1) and m(2); a picture follows:
%
% F-->m(1)--k(1)--m(2)--k(2)--m(3)--...
%
% System output y = [x1 act]'
%
% Where: act - force output of ideal actuator acting on mass 1;
% the transfer function of the actuator is 1/s.
% x1 - position of the mass being pushed by act (m[1]).
%
% If "gains" exists, then negative feedback of the states
% x1dot,x1,x2dot,x2,... is included in the state space model.
%
%% [PN,wn]=NORMP(P1)
% Normalizes a characteristic polynomial such
% that there are decending powers of S from the left and decending
% powers of W from the right (rightmost and leftmost coefficients are
% always unity. Returns the normalized polynomial and the WN used in
% the normalization. Polynomials should be entered such that the
% coefficient of the highest power of S is unity.
%
%% PRESR.M
% Step response for second order systems (res) and step response for
% second order systems with a three impulse prefilter (resp)
%
%% PRETF(Z,W,S) evaluates the transfer fuction of a three impulse prefilter
% designed for a system with the given z and w, at the given value of
% the complex variable s.
%
%% RAP2MZV(f1,f2,deltaT) -- Bill Singhose
% Generates a 2 mode ZV shaper based on
% Rappole equations for 2 undamped modes.
% Do not use this for frequency ratios over 3.76679 - it will
% result in gains larger than one.
% f1, f2 - frequencies(Hz) of vibration being controlled
% deltaT - time spacing at which input to system is updated
%
%% RAP2MZVD(f1,f2,deltaT) -- Bill Singhose
% Generates a 2 Mode ZVD shaper on based Rappole equations

```

```

% for 2 undamped modes.
% Do not use this for frequency ratios over 3.76679 - it will
% result in gains larger than one.
% f1, f2 - frequencies(Hz) of vibration being controlled
%
%% seq = Rapp9Seq(dt,f1,z1,f2,z2)
% Find an exact digital sequence found with Rappole 2-modes,
% 5-impulse constraints.
% Uses matrix inversion technique.
%
%% RESID.M
% residue analysis of dynamic systems
%
%% sensplot Plot the residual over range of frequencies
%
% list = sensplot(seq,fmin,fmax,zeta,points)
%
% seq is the shaping sequence
% fmin is the low end of the frequency range
% fmax is the high end of the frequency range
% zeta is the damping ratio of the system
% points is the number of points to calculate
%
%% SEQUENCECONVOLVE Convolve two continuous sequences together.
%
% seq = seqconv(seq1,seq2)
%
% Convolves two sequences together.
% A Sequence is an n*2 matrix with impulse times (sec) in
% the first column and amplitudes in the second column.
%
% Parameters:
% seq1, seq2 the two sequences to convolve together.
%
% Returns:
% seq, the sequence resulting from the convolution.
%
%% SEQUENCESORT Sort a continuous sequence into correct
% order and combine impulses at the same time.
%
% sortedseq = seqsort(unsortedseq)
%
% A sequence is an n*2 matrix with times (sec) in the first
% column and amplitudes in the second column.
%
% PARAMETERS:
% unsortedseq is the continuous sequence to sort
%
% RETURNS:
% sortedseq, the sorted sequence.
%
%% Shapers.m
% This file gives a brief description of the properties of the
% various shapers available in the Input Shaping Toolbox.

```

```

%
%% ShapGen --
% This programs Generates an Input Shaper
% from user supplied parameters
%
%% shapres Calculate the residual for a given freq and zeta
%
% num = shapres(seq,freq,zeta)
%
% seq is the sequence being used
% freq is the frequency of the system, Hz.
% zeta is the damping ratio
%
%% SPEEDCRM Speedy version of the controller redesign metric used
% to compute the residuals associated with residual vibration
% PRE - given prefilter
% T - sample period
% a,b,c,d - state space system description
%
% resy - vector of residuals for each Q2CxPole for output y.
% Q2CxPoles - vector of complex poles in the 2nd quadrant
%
% [resy,Q2CxPoles]=speedcrm(PRE,T,a,b,c,d)
%
% This function is on the fringe of numerical stability, speedy,
% and compact. The parameter delta is how far away from the
% actual pole that the "residual" is being evaluated. If we get
% too close to the pole, we have a 0/0 numerical situation; if
% we get too far away from the pole we no longer approximate the
% residual. The delta used here (1e-7) was determined by varing
% delta from 1E-2 to 1E-12 without a change in the printed
% output (0.0000 accuracy). 1e-7 is midway between the two
% values.
%
%% [T,P,V] = trapprof(endpos,curpos,amax,vmax,dt,steps)
%
% Whit Rappole
% Convolve, Inc.
% February 1, 1993
%
% c Copyright, 1993, Convolve, Inc.
% All Rights Reserved
%
% Generate position and velocity commands for a trapezoidal
% velocity profile.
%
% Parameters:
% endpos - Final position desired (in)
% curpos - Current position (in)
% amax - Maximum allowable acceleration (in/sec^2)
% vmax - Maximum allowable velocity (in/sec)
% dt - Time step for command string.
% steps - Total number of steps for command.
%

```

```

% Returns:
% T - Vector of times (sec)
% P - Vector of position commands (in)
% V - Vector of velocity commands (in/sec)
% T, P, and V are all of length STEPS
%
% If the desired motion requires more than STEPS time steps,
% TRAPEZOIDPROFILE returns [].
%
%%%TRAP_RUL
% trap_rul uses the trapezoidal rule on a vector and
% returns the area, since the spacing on the x-axis
% is dt, the "area" returned is the sum of all the trapezoids
%
%%% TWOMASS(w1,w2) returns a transfer function model for a two mass
% system with masses m1, m2 and a spring k between the masses.
%  $w1^2=k/m1$ ,  $w2^2=k/m2$ ; the natural frequencies that correspond
% to holding one mass fixed and twanging the other mass.
% The transfer function is for a force input to mass m1 with
% a displacement outputs at masses m1 and m2. (m1 case co-located)
% The transfer functions are for  $x1/(F/m1)$ ,  $x2/(F/m1)$ 
%
%%function vd(amp,time,freq,zeta)
% or vd(exact_shaper,freq,zeta)
% creates a vector diagram based on the
% provided f (natural frequency in hertz)
% and the times and amplitudes of the pulses
%
% returns the residual vibration x and y coordinates
%
% Mapping is ->  $\theta = w*time$ 
% So each vector amp[i],time[i] has a corresponding
% position in the vector diagram.
%
% Currently only handles positive vectors amp = [0 to 1]
%
%%% W_DIGSEQ Map a sequence onto digital timing loop
% dseq = W_DigSeq(seq,step,freq,zeta)
%
% Uses Watanabe's formula to split each continuous
% impulse into two digital impulses
% It is derived for Singer three impulse sequences.
%
% freq is the natural frequency in Hz.
% zeta is the damping ratio
% step is the time step of the timing loop, in sec.
% seq is the continuous sequence, an n x 2 matrix of
% times and amplitudes
%
%%% ZV(f,zeta,deltaT) -- Bill Singhose
% Generates a ZV shaper for 1 mode.
% f - frequency (Hz) of vibration being controlled.
% zeta - damping ratio of vibration being controlled.
% deltaT - time spacing at which input to system is updated.

```

```

%
%% ZVD-2M -- Bill Singhose
% Generates a ZVD shaper for 2 modes.
% The shapers for each mode are convolved together to give the
% final shaper.
% f1,f2 - frequencies (Hz) of vibration being controlled.
% zeta1,zeta2 - damping ratios of vibration being controlled.
% deltaT - time spacing at which input to system is updated.
%
% Check to see if there is sufficient frequency seperation to
% justify using this filter.
%
%% ZVD(f,zeta,deltaT)-- Bill Singhose
% Generates a ZVD shaper for 1 mode.
%
% Returns the digital sequence by default, and
% the exact sequence as the second variable:
% eg: to generate the exact shaper, use:
%           [dig_shaper,exact_shaper]=singer13(f,z,0.01)
%           note: the 0.01 is ignored for the exact shaper and
%           dig_shaper can be ignored
%
% to Return the digital sequence, use:
%           dig_shaper=singer13(f,z,0.001)
%
% f - frequency (Hz) of vibration being controlled.
% zeta - damping ratio of vibration being controlled.
% deltaT - time spacing at which input to system is updated.
%
%% ZVD2MD(f1,f2,zeta1,zeta2,deltaT) -- Bill Singhose
% Generates a damped 2 mode ZVD shaper
% The solution presented here is the result of a curve fit to data
% generated by GAMS.
% This shaper works well over the ranges of  $1.2 < r < 2.8$  and  $0 < z < .15$ 
% fd1, fd2 - frequencies(Hz) of vibration being controlled
% zetas - damping ratios of low mode
% deltaT - time spacing at which input to system is updated
%
%% ZVDD -- Bill Singhose
% Generates a ZVDD shaper for 1 mode.
% f - frequency (Hz) of vibration being controlled.
% zeta - damping ratio of vibration being controlled.
% deltaT - time spacing at which input to system is updated.
%
% This function generates the exact sequence and then uses
% DigitizeSeq to convert the exact sequence to digital format.
%
%% only for ZERO DAMPING RIGHT NOW!!
% ZVDDD -- Bill Singhose
% Generates a ZVDDD shaper for 1 mode.
% f - frequency (Hz) of vibration being controlled.
% zeta - damping ratio of vibration being controlled.
% deltaT - time spacing at which input to system is updated.
%

```


REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1995	3. REPORT TYPE AND DATES COVERED Contractor Report	
4. TITLE AND SUBTITLE Minimizing Structural Vibrations with Input Shaping™			5. FUNDING NUMBERS Code 714.1 NAS5-32034	
6. AUTHOR(S) Bill Singhose & Neil Singer				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Convolve, Inc. One Quarter Mile Road Armonk, NY 10504			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) NASA Aeronautics and Space Administration Washington, D.C. 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER CR-189439	
11. SUPPLEMENTARY NOTES Technical Monitor: H. Frisch/714.1				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category: 39 Report available from the NASA Center for AeroSpace Information, 800 Elkridge Landing Road, Linthicum Heights, MD 21090; (301) 621-0390.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) A new method for commanding machines to move with increased dynamic performance was developed. This method is an enhanced version of Input Shaping™, a patented vibration suppression algorithm. This technique intercepts a command input to a system command that moves the mechanical system with increased performance and reduced residual vibration. This document describes many advanced methods for generating highly optimized Shaping Sequences™ which are tuned to particular systems. The Shaping Sequence™ is important because it determines the trade off between move/settle time of the system and the insensitivity of the Input Shaping™ algorithm to variations or uncertainties in the machine which can be controlled. For example, a system with a 5 hz resonance that takes 1 second to settle can be improved to settle instantaneously using a 0.2 Shaping Sequence™ (thus improving settle time by a factor of 5). This system could vary by ± 15% in its natural frequency and still have no apparent vibration. However, the same system shaped with a 0.3 second Shaping Sequence™ could tolerate ± 40% or more variation in natural frequency. This document describes how to generate sequences that maximize performance, sequences that maximize insensitivity, and sequences that trade off between the two. Several software tools are documented and included with this report to the Principal Investigator at NASA.				
14. SUBJECT TERMS Vibration Reduction, Vibration Suppression, Settling, Input Shaping, Command Shaping, Slewing			15. NUMBER OF PAGES 160	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	