Submitted to the 1991 IEEE International Conference on Robotics and Automation 321

N93-25834

N93

51-31

p. 19

Using Input Command Pre-Shaping to Suppress Multiple Mode Vibration

James M. Hyde
Research Assistant
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Warren P. Seering
Professor
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

Spacecraft, space-borne robotic systems, and manufacturing equipment often utilize lightweight materials and configurations that give rise to vibration problems. Prior research has led to the development of input command pre-shapers that can significantly reduce residual vibration. These shapers exhibit marked insensitivity to errors in natural frequency estimates and can be combined to minimize vibration at more than one frequency. This paper presents a method for the development of multiple mode input shapers which are simpler to implement than previous designs and produce smaller system response delays. The new technique involves the solution of a group of simultaneous non-linear impulse constraint equations. The resulting shapers were tested on a model of MACE, an MIT/NASA experimental flexible structure.

Introduction

Space-borne robotic systems and vehicles often employ lightweight materials and configurations that result in a high degree of system flexibility. The system's light weight facilitates launching, but chronic vibration problems are a common result. Manufacturing equipment also increasingly utilizes lighter structural elements, a main objective being improving the speed of automated assembly. The combination of a lightweight structure with high performance requirements often leads to serious vibration problems. The growing demand for high accuracy manipulation is in no way aided by these simultaneous attempts to increase speed and decrease weight. Partial or complete suppression of system vibration can improve spacecraft durability and performance, and would allow manufacturing systems to operate faster and more economically.

Attempts to decrease the vibration inherent in flexible systems have enjoyed varied success over the past decade. Cannon and Schmitz [1] experimented with the non-colocated feedback control of a flexible beam. Through the use of accurate system models and optical tip position sensing they achieved significant vibration reduction in their planar test article.

Yurkovich and Tzes [8] reduced vibration in the presence of unknown and/or varying payloads by employing on line system identification and controller tuning. By using frequency domain techniques to examine the system response following a sample input, enough information was gained to adjust the controller gain scheduling to compensate for vibration problems.

Wie [9] employed H_∞ controllers to reduce vibration while providing robustness to modelling errors. This technique displayed solid performance, but was relatively difficult to implement.

Input command shaping is an attractive vibration reduction method because it is essentially "hands off;" inputs can be fed through a shaper and into the system, and ideally the resulting system output will be vibration free. Shapers also usually reside completely outside of a given control system and are thus easily compatible with other vibration schemes (see figure 1). Smith

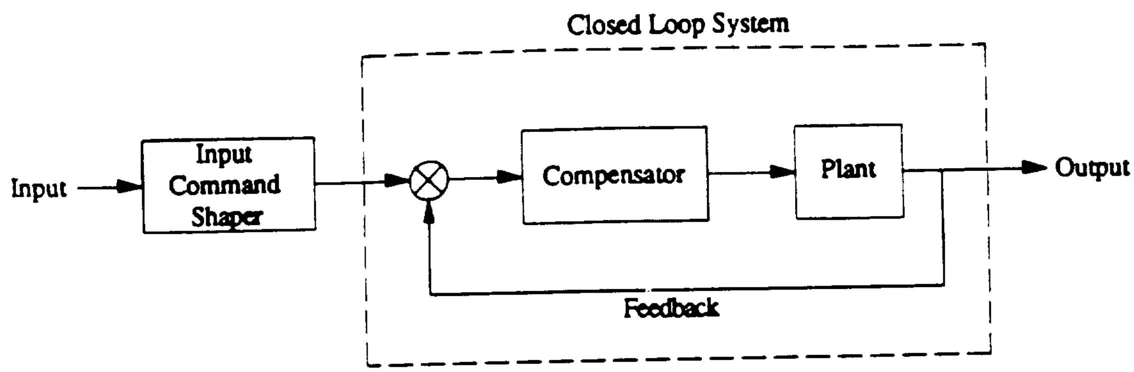

Figure 1: Shaper Position in Control System

[6] conducted early shaping investigations, largely through the use of posicast control.

Meckl [3] examined the use of shaped force profiles to reduce vibration in manufacturing systems. Meckl created profiles by using a versine (1 – cosine) function to modify force commands. When integrated twice, these force profiles became input trajectories that reduced system vibration at a structure's first natural frequency.

A major problem with command shaping is that its success usually depends on solid prior knowledge of plant dynamics. Many attempts at input shaping have been criticized because the shapers exhibited significant dependence on precise system models.

Singer [4] presented a simple shaping algorithm that demonstrated strong insensitivity to modelling errors. The shapers were assembled from impulse sequences and produced only small delays in system response times, on the order of one period of a system's natural frequency. This technique performed notable vibration reduction in tests of a full scale mockup of the Space Shuttle Robotic Manipulator System, conducted at NASA's Manipulator Development Facility at the Johnson Space Flight Center.

Tzes, Englehart, and Yurkovich [7] studied the effects of combining input shaping with a closed loop, acceleration feedback controller. They conducted tests on a flexible beam and proved that each technique can complement the other, resulting in enhanced vibration reduction. This work supports the assertion that input command shaping can be used concurrently with other vibration suppression schemes.

Singer [4] originally assembled shapers designed to cancel single mode vibration and later expanded the algorithm to handle multiple mode problems. The initial multiple mode technique was somewhat cumbersome, however, and the main purpose of this paper is to present an improved method for developing multiple mode shapers. Simpler impulse trains can be assembled by directly solving a full set of multiple mode vibration equations. These new shapers have all the vibration reduction capabilities of the original shapers, and yet exhibit savings in implementation complexity and response time. We present an approach for solving the vibration equations and offer evidence of the new shapers' potential through tests conducted on a model of MACE, an MIT/NASA experimental flexible structure.

Single Mode Shaping

To develop a single mode input shaper, we first note that the second order system response to an impulse input is described by:

$$y_i(t) = A_i\, e^{-\zeta\omega(t-t_i)} \sin\left((t-t_i)\omega\sqrt{1-\zeta^2}\right) \tag{1}$$

where $y_i(t)$ is the output, A_i is the impulse amplitude and t_i is the time at which the impulse occurs. The system's vibration frequency is ω, with damping ζ. If the system is linear, its total response to a series of N impulses can be expressed as a sum of the the responses to each impulse "i." The magnitude of the total response following the Nth impulse is given by:

$$\mathrm{Amp} = \left[\left(\sum_{i=1}^{N} A_i \, e^{-\zeta\omega(t_N - t_i)} \sin(t_i \omega \sqrt{1-\zeta^2}) \right)^2 + \left(\sum_{i=1}^{N} A_i \, e^{-\zeta\omega(t_N - t_i)} \cos(t_i \omega \sqrt{1-\zeta^2}) \right)^2 \right]^{1/2} \tag{2}$$

A train of properly arranged impulses can suppress residual vibration by forcing Amp to equal zero. This can only happen when both the sine and cosine terms in equation (2) independently equal zero:

$$\begin{aligned} \sum_{i=1}^{N} A_i \, e^{\zeta\omega t_i} \sin(t_i \omega \sqrt{1-\zeta^2}) = 0 \\ \sum_{i=1}^{N} A_i \, e^{\zeta\omega t_i} \cos(t_i \omega \sqrt{1-\zeta^2}) = 0 \end{aligned} \tag{3}$$

To construct an impulse sequence that will act as a vibration reducing input shaper, we start by imposing two initial constraints:

$$t_1 = 0 \tag{4}$$

$$\sum_{i=1}^{N} A_i = 1 \tag{5}$$

The first is simply an origin specification, and the second is a normalization constraint. Normalizing a shaper's impulse magnitudes ensures that a shaped input will not exceed limitations imposed on the original input, such as actuator or stress limits. We specify an arbitrary value for A_1, and with N = 2, we can use equations (3) to solve for the time and amplitude of the second impulse in a two-impulse shaper.

This shaper will completely cancel residual vibration in a single mode system, as long as the natural frequency and damping ratio are perfectly known. To account for possible modelling inaccuracies, the shaper should exhibit some insensitivity to errors in natural frequency and damping ratio estimates. By differentiating equations (3) with respect to natural frequency, we generate two additional impulse constraints:

$$\sum_{i=1}^{N} A_i\, t_i\, e^{\zeta\omega t_i} \sin\!\left(t_i\omega\sqrt{1-\zeta^2}\right) = 0$$
$$\sum_{i=1}^{N} A_i\, t_i\, e^{\zeta\omega t_i} \cos\!\left(t_i\omega\sqrt{1-\zeta^2}\right) = 0 \qquad (6)$$

Setting the partial derivative with respect to natural frequency equal to zero also sets the partial derivative with respect to damping ratio equal to zero [4]. These new constraints require the addition of a third impulse to our sequence; we have four equations, we need two unknown amplitudes and two unknown times. The three impulse sequence will force the residual vibration to be low even if the system parameters are not precisely known.

The standard three impulse, single mode shaper features impulses with a 1-2-1 magnitude configuration and times that are equally spaced. A typical sequence is shown in figure 2.

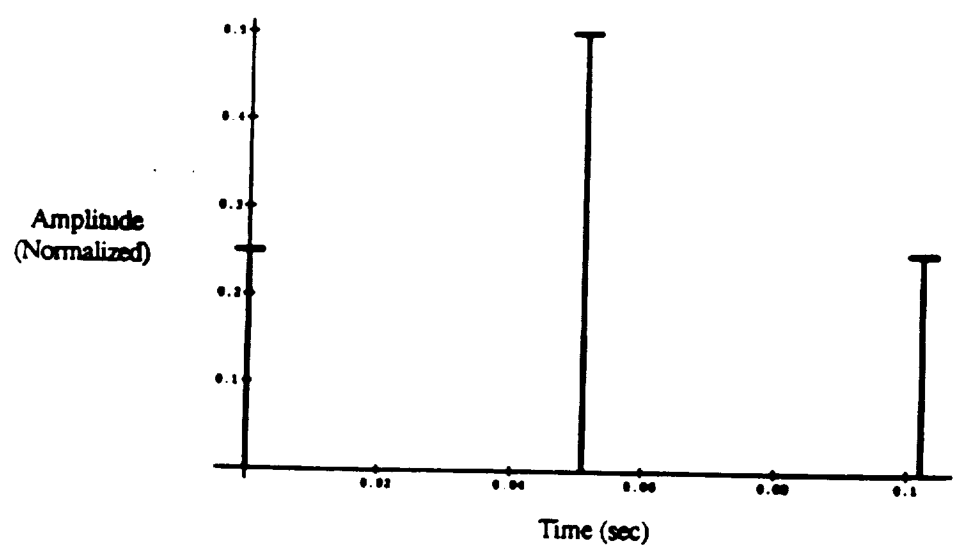

Figure 2: Typical single mode three impulse shaper.

Note: these impulses are constrained to having positive amplitudes. By using negative impulses, the time of a series' final impulse can be decreased, but negative impulses tend to tax a system's actuators and introduce high stress levels. In the remainder of this paper, all shapers will utilize impulses with positive amplitudes. For a full derivation of the above equations, see Singer [4].

Adding Modes

To cancel multiple mode vibration, we can convolve several single mode impulse sequences into longer trains. Convolution results in a sequence whose final impulse is located at a time equal to the sum of the damped periods of the cancelled modes. The value of the final impulse's time will be referred to as the shaper's "length." The number of impulses in the convolved sequence is equal to 3^m, where m equals the number of modes. A standard three mode convolved shaper is shown in figure 3. This sequence was solved for a zero damping case, so the twenty-seven impulses are arranged symmetrically about the center of the pattern.

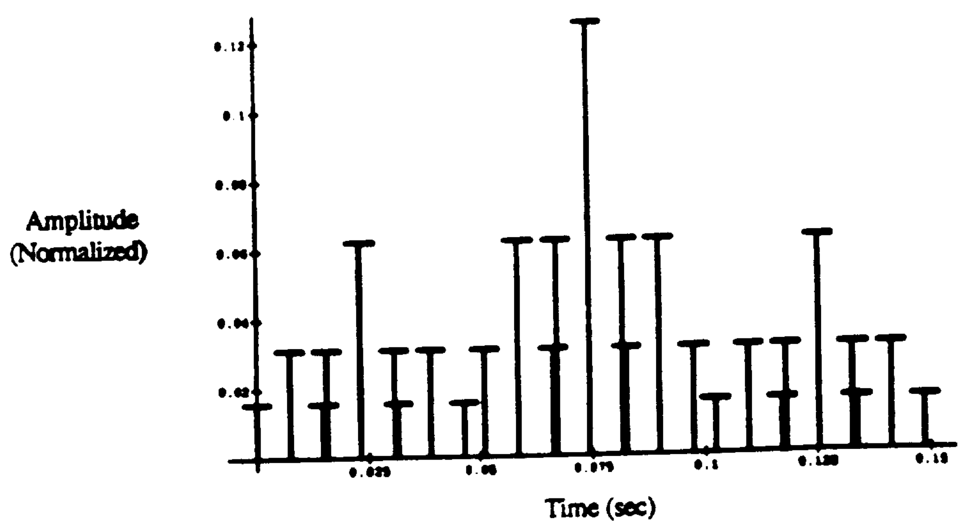

Figure 3: Three mode convolved impulse sequence.

The convolved multiple mode sequences are easily generated, but their failings become clear when the cancellation of higher mode vibration is desired. The number of impulses in a convolved shaper increases exponentially with added modes, and by the time the third or fourth mode is added, the sequence has become packed with impulses and can be difficult to implement in real time. Shapers with more impulses increase the time required to modify an input, and might force a decrease in servo rate.

The solution to these problems is to build multiple mode sequences not through convolution of single mode sequences, but through a direct solution of the constraint equations, (3) and (6), as written to include an arbitrary number of modes:

$$\sum_{i=1}^{N} A_i e^{\zeta_j \omega_j t_i} \sin(t_i \omega_j \sqrt{1-\zeta_j^2}) = 0$$
$$\sum_{i=1}^{N} A_i e^{\zeta_j \omega_j t_i} \cos(t_i \omega_j \sqrt{1-\zeta_j^2}) = 0 \qquad (7a)$$

$$\sum_{i=1}^{N} A_i t_i e^{\zeta_j \omega_j t_i} \sin(t_i \omega_j \sqrt{1-\zeta_j^2}) = 0$$
$$\sum_{i=1}^{N} A_i t_i e^{\zeta_j \omega_j t_i} \cos(t_i \omega_j \sqrt{1-\zeta_j^2}) = 0 \qquad (7b)$$

Repeating equations (7) for additional modes "j" generates a set of simultaneous non-linear impulse expressions. Solving these equations can yield shapers with shorter lengths than the convolved shapers. Shorter sequence lengths decrease the delay in system response caused by using the shaper. The direct solution sequences, moreover, use only (2*m) + 1 impulses, m being the number of cancelled modes. This linearly increasing impulse population leads to vastly fewer impulses in higher mode shapers, reducing implementation time.

The savings in length and impulse density that the direct solution sequences support are offset by an increase in sequence generation complexity. The single mode shaper equations, given the constraints of positive impulse amplitudes and shortest possible overall length, had a closed form solution. As written for the multiple mode case, the shaper equations require a strict set of constraints just to limit their infinite solution space, and no general solution has been found.

Solving the Equations

The key to solving the multiple mode equations thus far has been to employ a linear approximation. Equations (7) are non-linear only in terms of impulse time. A straightforward approach is to pick a time for the sequence's final impulse, essentially defining a sequence length, and then divide the length into a fine time mesh. An impulse is placed at each time slot, with unknown amplitude but known time. The equations are now under constrained, but a linear approximation to the exact shaper sequence can be generated through optimization.

The constraints for the optimization problem are the multiple mode equations (7) and the normalization requirement (5). The cost function is the sum of the second derivatives of equations (7a):

$$\text{Cost} = \Big| \sum_{j=1}^{M} \sum_{i=1}^{N} A_i t_i^2 e^{\zeta_j \omega_j t_i} \sin(t_i \omega_j \sqrt{1 - \zeta_j^2}) + \sum_{j=1}^{M} \sum_{i=1}^{N} A_i t_i^2 e^{\zeta_j \omega_j t_i} \cos(t_i \omega_j \sqrt{1 - \zeta_j^2}) \Big| \tag{8}$$

Minimizing the second derivative expressions forces the impulse sequence to be even more insensitive to modelling errors. With these guidelines, the linear problem becomes "minimize the cost function subject to the stated constraints."

Solution of the linear problem was accomplished using GAMS, a standard optimization package. GAMS utilizes a version of the primal simplex method to perform linear optimization. If M is the number of modes to be cancelled, σ is the length of the sequence, and dt is the value of a single time mesh element, GAMS' constraint matrix consists of:

$$\text{rows: } r = (4 * M) + 1$$

$$\text{columns: } c = \sigma / dt$$

The variable vector is the series of impulse amplitudes, A_i. The simplex method dictates that at least (c - r) amplitudes will equal zero, and additional impulses are occasionally set with zero amplitude. The optimized GAMS output yielded an impulse train with a number of impulses that was less than or equal to r. This train was a linear approximation to the exact multiple mode shaper.

The second phase of the linear work was to find the feasible solution with the smallest possible final impulse time. This was achieved through multiple GAMS runs, systematically reducing the time of the final impulse and using a binary search algorithm that recognized when GAMS returned an infeasible solution, meaning that the time had been reduced too far. This technique ensured that the final GAMS output was the shortest possible approximation. Figure 4 shows a typical final GAMS result from a three mode problem.

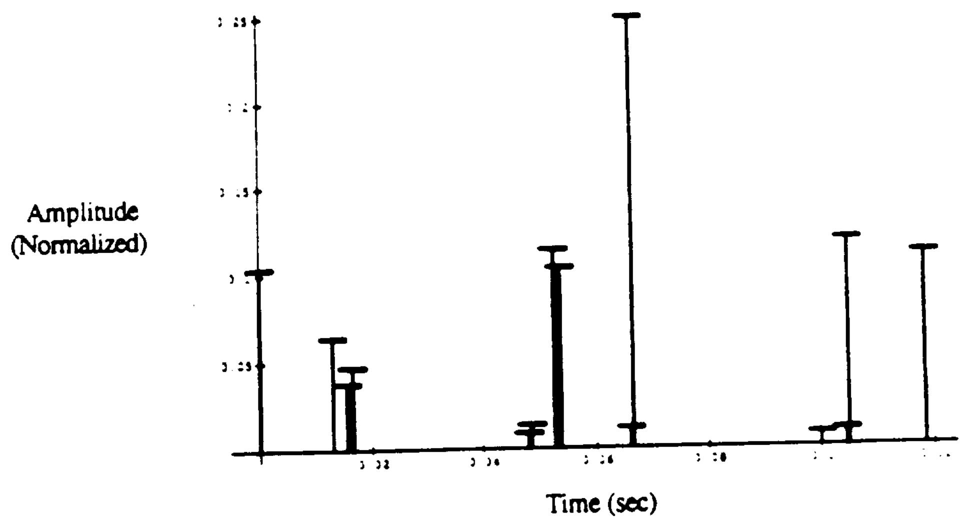

Figure 4: GAMS output for a standard three mode system.

In many cases, this output could stand alone as an effective shaper sequence, especially if the time mesh was set at a digital controller's servo rate. The raw GAMS output, however, had about twice as many impulses as the exact solution demanded. Our goal was to arrive at a sequence with as few impulses as possible, so we used the GAMS output as an initial guess in a non-linear equation solver.

Given the large number of impulses in the final approximate sequence, the GAMS output had to be interpreted to obtain useful guesses of the exact solutions to equations (7). GAMS would often place impulses in adjacent time intervals; these impulses were replaced by single spikes that combined the amplitudes of the neighbors and adopted their exact average time. To further reduce the number of impulses, the interpretation algorithm sought out the closest non-adjacent neighbors. These pairs were combined by summing their amplitudes and taking a weighted average of their times. This set of techniques yielded a sequence whose number of impulses matched that required by the non-linear multiple mode shaper equations (7).

Mathematica™ was used as the non-linear equation solver, mainly because of its additional potential as a programming language that could envelop the entire computational side of the impulse sequence generation. The time and amplitude of the first ($i = 1$) impulse in equations (7) were held constant, matching the first impulse from the interpreted GAMS output. The remaining times and amplitudes were allowed to vary, with initial guesses of their values provided by the reduced GAMS sequence.

Mathematica employs a Newtonian gradient search algorithm to arrive at its solutions. This method worked quite well, as long as our guesses were sufficiently close to the optimal solutions. As the non-linear equations are continuous and differentiable, adequate gradients are readily available, and points of singularity are usually easy to avoid. The resultant exact impulse sequence, after interpreting the GAMS output in Figure 4 to find initial guesses, is shown in Figure 5. The impulses in the Mathematica result were re-normalized to ensure that the constraint of equation (5) was upheld.

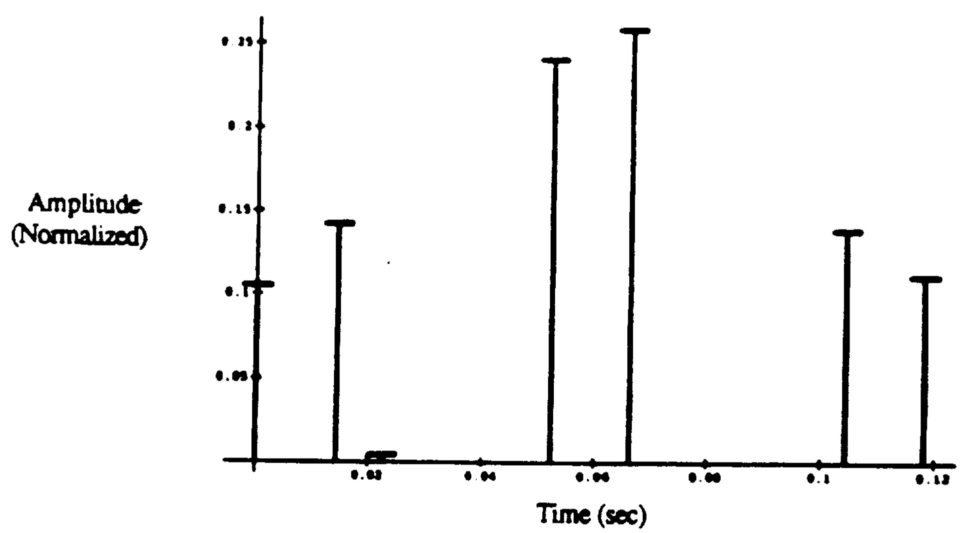

Figure 5: Exact impulse sequence solution, output from Mathematica.

We found that Mathematica could solve the equations only part of the time. A particular structure's absolute modal values and relative modal spacing could disrupt the GAMS program, or Mathematica, or both, resulting in a group of unsolved equations. A variety of different approaches have been used to increase the robustness of the solution process, the ultimate goal being the discovery of a closed form solution. While current work is continuing in this area, the linear approximation/non-linear solution algorithm has successfully generated workable impulse sequences for different groups of three and five modes.

Modeling and Results

The three mode case of particular interest involves a set of frequencies found using a model of an actual flexible system, the MACE test article. The MACE experiment is a joint MIT/NASA project designed to study methods for controlling flexible systems in micro or zero gravity fields. MACE is a flexible structure with two multi-axis pointing payloads residing on either end of a tubular bus. The system incorporates attitude control through a set of three-axis torque wheels, and utilizes inertial position sensing information gained from gyroscope packages mounted at the center of the bus and inside each payload. A simple system schematic is shown in Figure 6.

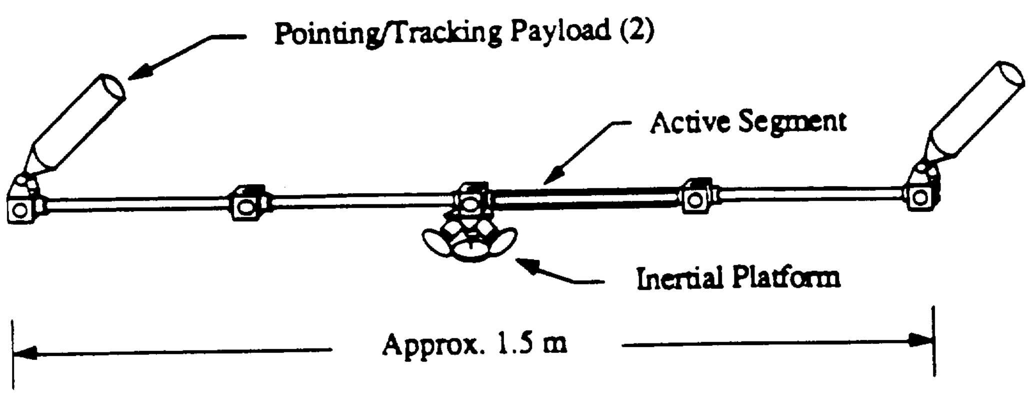

Figure 6: The MACE flexible test article.

The MACE project consists of a Ground Test Article, to reside at MIT, and a Flight Test Article, scheduled for a Space Shuttle launch in 1993. The Ground Test Article is currently being assembled and should be available for testing in December of 1990. The ground article will be actively suspended to emulate a flexible spacecraft in a micro gravity field. This experiment is a prime candidate for practical validation of the command shaper techniques.

While the physical MACE structure has been under construction, personnel at MIT's Space Engineering Research Center (SERC) have developed several computer models of the experiment. The frequencies used in the three mode case mentioned above were found using a linear finite element MatLab model of MACE. The planar model depicted the segmented bus and one of the pointing payloads, as shown in Figure 7.

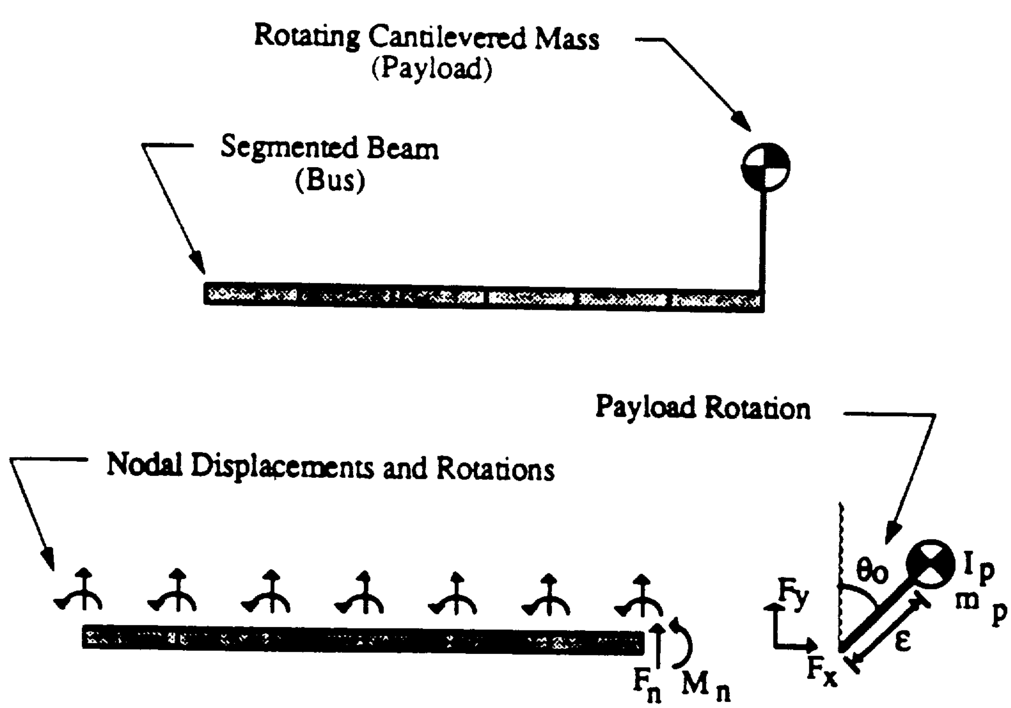

Figure 7: Finite element model of MACE.

The first three eigenvalues of the model were fed into the GAMS / Mathematica routine, generating the shaper sequence shown in Figure 5. Next, simple torque inputs were fed through the shapers and into the modelled payload's gimbal axis, producing the adjusted inputs shown in figure 8. The resulting translation of the beam element on the opposite end of the bus is shown in figure 9, and detailed views of the unshaped and shaped response are provided by figures 10 and 11, respectively. The model had a system of eight modes of vibration, and only the first three were used in forming the input shaper. It is clear from the figures, however, that cancelling these three modes was sufficient to suppress the majority of the structure's vibration.

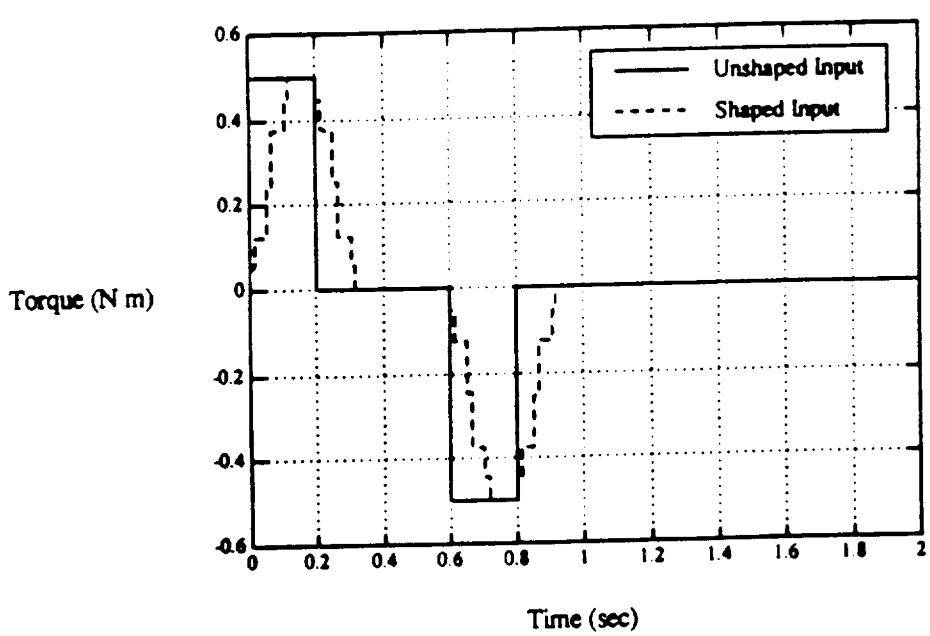

Figure 8: System inputs adjusted by the input shaper.

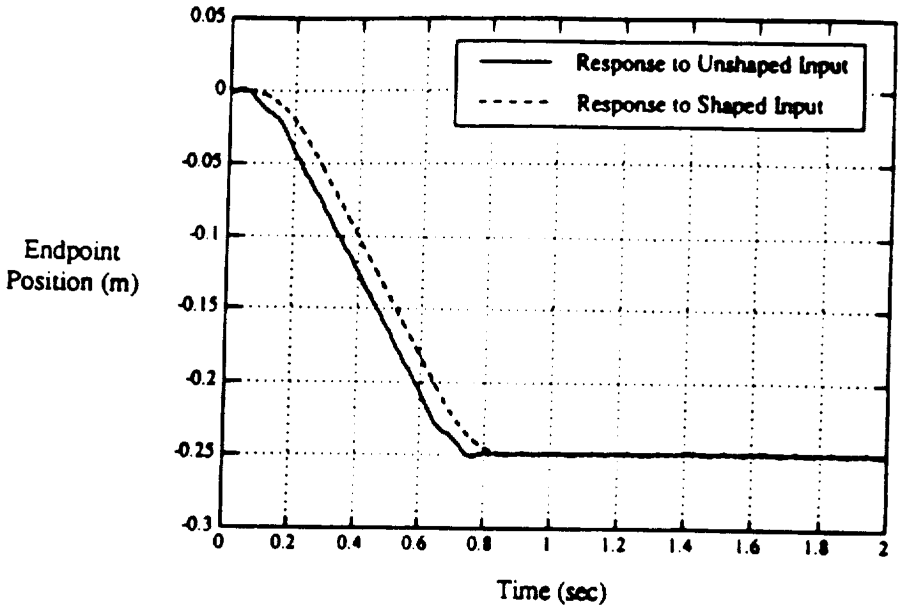

Figure 9: System response to inputs.

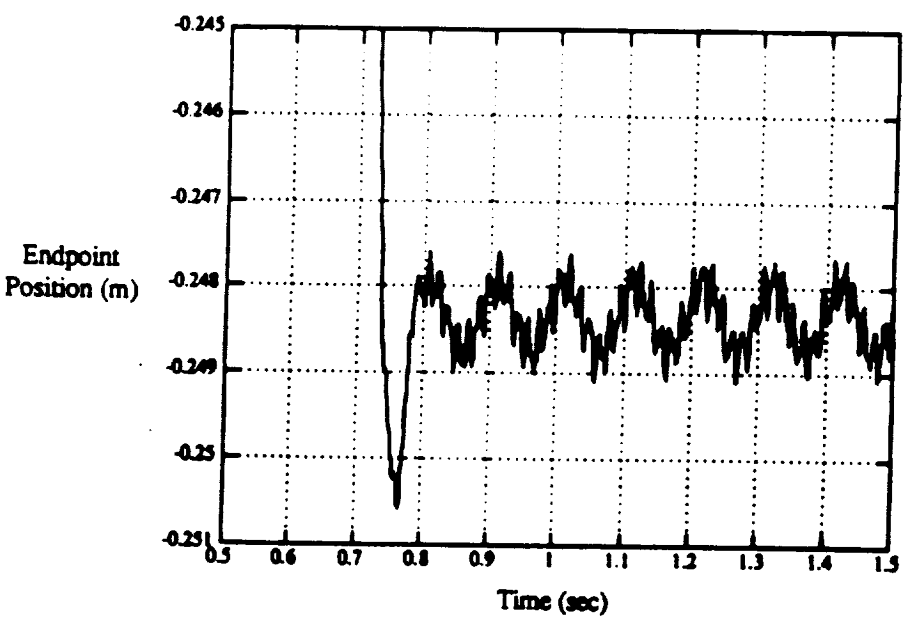

Figure 10: Response to unshaped input (detail).

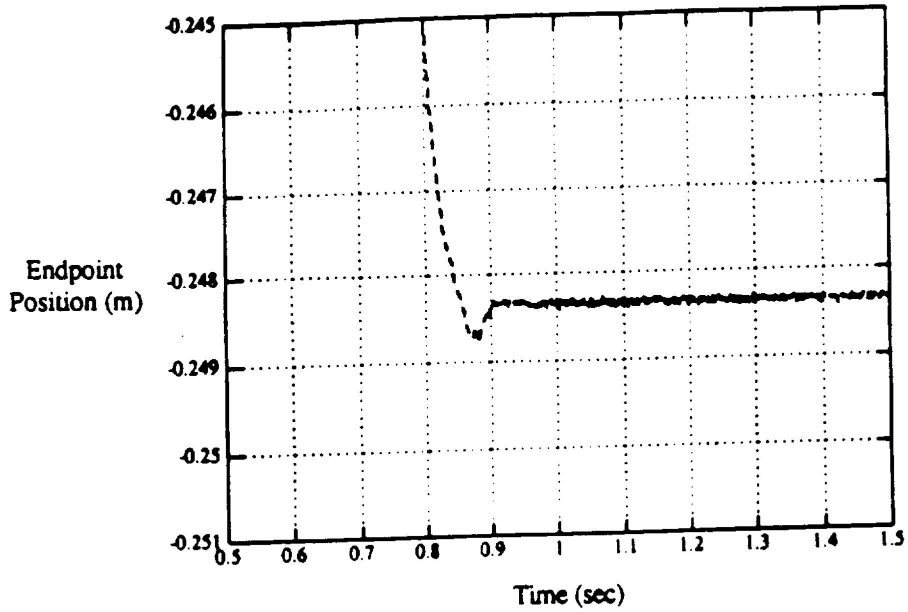

Figure 11: Response to shaped input (detail).

Conclusions

These MatLab results are somewhat predictable. The input shapers are defined by equations that predict the response of linear systems, and the MatLab model was also linear. Cancelling the vibrations of the MatLab model, therefore, served mainly as a confirmation of the proper solution of the constraint equations, and allowed for concrete visualization of what a system experiences when the input shapers are employed. The lessons learned from this initial case will also be valuable when more complex, higher mode shapers are developed.

The next step in this program is to employ more accurate models of MACE. The test article has been simulated non-linearly, using the DISCOS program. This model will likely predict some of the shaper's failings in suppressing vibration in non-linear systems.

The second major future task is to improve the equation solving algorithm to facilitate the construction of higher mode shapers. Sequences that can cancel up to ten or fifteen modes are not out of the question. In addition to increasing the number of cancelled modes, we are devoting effort to decreasing the sequence generation time. We have shown that the direct solution sequences are easier to implement than the convolved sequences, but they are much more difficult to generate. These continued efforts to solve the constraint equations, coupled with the lessons learned from the DISCOS model, will aid in the generation of input shapers capable of effectively reducing vibrations in the actual MACE structure.

Acknowledgements

This paper describes research performed at the Massachusetts Institute of Technology Artificial Intelligence Laboratory and Space Engineering Research Center. Funding for this work was provided in part by the National Aeronautics and Space Administration under grant #NAGW-1335. Additional support was provided by the Office of Naval Research under the University Research Initiative contract #N00014-86-K-0685.

References

[1] **Cannon, Robert H., and Schmitz, Eric,**
"Initial Experiments on the End-Point Control of a Flexible One-Link Robot," The International Journal of Robotics Research, Vol. 3, No.3, Fall 1984.

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

[3] **Meckl, Peter H. and Seering, Warren P.,**
"Controlling Velocity-Limited Systems to Reduce Residual Vibration," IEEE Conference on Robotics and Automation, April 1988.

[4] **Singer, Neil C.,**
"Residual Vibration Reduction in Computer Controlled Machines," MIT Artificial Intelligence Laboratory Technical Report #1030, February 1989.

[5] **Singhose, William E.,**
"Shaping Inputs to Reduce Residual Vibration: A Vector Diagram Approach," MIT Bachelor of Science Thesis, February 1990.

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

[7] **Tzes, A. P., Englehart, M. J., and Yurkovich, S.,**
"Input Preshaping With Frequency Domain Information For Flexible-Link Manipulator Control," AIAA Guidance, Navigation, and Control Conference, August 1989.

[8] **Yurkovich, S., Pacheco, F. E., and Tzes, A. P.,**
"On-Line Frequency Domain Information for Control of a Flexible-Link robot with Varying Payload," Ohio State University Control Research Laboratory Report #CRL-1036-Su88-P, 1989.

[9] **Wie, Bong, and Liu, Qiang,**
"Feedforward/Feedback Control Synthesis for Performance and Robustness," AIAA Guidance, Navigation, and Control Conference, 1990.