

**IDENTIFICATION OF NATURAL FREQUENCY COMPONENTS OF
ARTICULATED FLEXIBLE STRUCTURES**

A Thesis
Presented to
The Academic Faculty

by

Dewey Hobson Lane III

In Partial Fulfillment
of the Requirements for the Degree
Master of Science from the Mechanical Engineering Department

Georgia Institute of Technology
October, 1996

**IDENTIFICATION OF NATURAL FREQUENCY COMPONENTS OF
ARTICULATED FLEXIBLE STRUCTURES**

Approved:

Wayne D. Book, Chairman

Aldo A. Ferri

Kok-Meng Lee

Date Approved 10/18/96

*Dedicated to
my family in appreciation of their lifelong encouragement and support*

ACKNOWLEDGMENTS

The author wishes to acknowledge the support and guidance of Dr. Wayne J. Book whose candid advice and continuous encouragement made this project possible. The author would also like to thank Dr. Aldo Ferri for his enthusiastic and thorough presentation of course-work relevant to this project as well as his input as a committee member during the writing of this thesis. Finally, I would like to thank Dr. Kok-Meng Lee for serving on this committee.

The members of the author's research group, David Magee, David Cannon, Sungsoo Rhim, Jonathan Hogan, Klaus Obergfell, Paul Schmidtt, Andy Register, and Hurley Davis provided insight, friendship, and humorous diversion above and beyond the call of duty.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	iv
TABLE OF CONTENTS	v
LIST OF TABLES	ix
LIST OF FIGURES.....	x
GLOSSARY OF ABREVIATIONS.....	xii
GLOSSARY OF NOTATION	xv
SUMMARY	xvi
CHAPTER	
I. INTRODUCTION.....	1
1.1 Adaptive Control Motivation	2
1.1.1 Manipulator Vibration Reduction Problem.....	2
1.1.2 Hardware Design Solutions	3
1.1.3 Controller Design Solutions.....	4
1.1.4 Adaptive Control Solutions	6
1.2 Prior Work in System Identification for Adaptive Control	6
1.2.1 Fundamentals of Frequency Domain System Identification, ETFE.....	8
1.2.2 Time-varying Transfer Function Estimation, TTFE	9
1.2.3 Real-Time Application of TTFE	12
1.2.4 Parameterization of the TTFE	14

1.3 Prior Work in Control of Vibration in Mechanical Structures	15
1.3.1 Flexible Dynamics Control.....	15
1.3.2 Adaptive Control for Vibration Reduction	16
II. FREQUENCY DOMAIN SYSTEM IDENTIFICATION.....	19
2.1 Introduction.....	19
2.1.1 Problem Statement	19
2.2 Dynamic Model of Testbed	21
2.2.1 State Space Representation	22
2.2.2 Transfer Function Representation	23
2.3 System Identification Algorithms.....	25
2.3.1 Recursive Least Squares.....	25
2.3.2 TTFE	26
2.3.3 Full Width Half Maximum, FWHM	27
2.4 Modifications to TTFE.....	28
2.4.1 Frequency Domain Smoothing.....	28
2.4.2 Hybrid Time-Frequency Domain Smoothing	28
2.4.3 Multiple Input, Multiple Output Modification.....	30
2.4.4 Summary of Modifications.....	36
III. HARDWARE AND IMPLEMENTATION	37
3.1 RALF, Robot Arm, Large and Flexible.....	37
3.1.1 Sensors	39

3.1.2 Actuators	40
3.2 Computation and Control.....	40
3.2.1 Hardware.....	41
3.2.2 Software	42
3.2.3 Handshaking and Buffering.....	46
3.3 Identification and OATF Filtering during Feedback Control	49
3.3.1 Location of OATF Filter and Identifier in Feedback Control System.....	49
3.3.2 Discrete OATF Time-Delays	51
IV. RESULTS.....	55
4.1 DSP Implementation Verification.....	55
4.1.1 TFE Implementation Verification.....	55
4.1.2 Algorithm Verification.....	56
4.2 Experimental Performance Comparison of Adaptive vs. Fixed OATF Filtering ..	58
4.2.1 Performance Evaluation Criteria	60
4.2.2 PI Controller	61
4.2.3 Performance Comparison for Various Trajectories and Payloads.....	65
4.3 Simulation.....	69
4.3.1 Model	69
4.3.2 Results	70
4.4 Reduced Damping OATF Filtering Experiment	74
V. CONCLUSION.....	78

5.1 Evaluation of Results	78
5.1.1 Poor Adaptive OATF Performance.....	78
5.1.2 Possible Advantages of Adaptive OATF Filtering	79
5.2 Contributions of This Research	80
5.3 Future Work	81
5.4 Conclusion.....	82
APPENDIX	
A. KINEMATICS OF RALF.....	83
A.1.1 Forward Kinematics.....	83
A.1.2 Inverse Kinematics	84
B. FULL WIDTH HALF MAXIMUM DERIVATION.....	87
C. VARIABLE RESIDUE METHOD OF POLE PARAMETERIZATION	92
D. RECURSIVE LEAST SQUARES.....	98
E. ADAPTIVE PIECEWISE POLYNOMIAL REGRESSION.....	108
BIBLIOGRAPHY	112

LIST OF TABLES

Number	Page
4-1 RMS Velocity and Acceleration Over Entire Trajectory, Adaptive vs. Fixed OATF	68
4-2 Simulation: Performance Differential Gained by Tuning OATF	71
A-1 Denavit-Hartenberg Parameters for RALF	83

LIST OF FIGURES

Number	Page
1.1 Second Order System With a Vibration Absorber Installed	4
2.1 Two Input, Two Output, Transfer Function Model	31
3.1 RALF, Robot Arm, Large and Flexible	38
3.2 Software Module Distribution and Interaction	42
3.3 C++ Class Hierarchy and Interaction	44
3.4 Performance Comparison of Hardware and Software	46
3.5 DSP-CPU Handshaking Schematic	48
3.6 Possible OATF and Identifier Locations in Feedback Control of RALF	50
3.7 Improperly Designed OATF, Frequency Response	53
3.8 Correct Discrete Approximation OATF , Frequency Response	53
4.1 DSP Transfer Function Estimation Code Verification	56
4.2 DSP Transfer Function Estimate Verification	58
4.3 Desired Tip Trajectory	59
4.4 Link Deflection Under PI Control	62
4.5 Spectrogram of PI Control Experiment	64
4.6 Matlab Spectrogram of Link 2 Deflection Data	65
4.7 Simulation: Performance Differential Obtained by Tuning OATF	72
4.8 OATF Frequency Response for Reduced Damping Ratio	73
4.9 Desired Trajectory and Filtered Trajectory Plus Noise	74

4.10	Deflection Measurements During Adaptive Filtering Near Singularity	76
4.11	Deflection Measurements Over Entire Trajectory While Adaptive Filtering	77
A.1	Kinematics Diagram of RALF	84
B.1	FWHM Calculation Error	91
C.1	Empirical Transfer Function Simulation	94
C.2	Natural Frequency Computed From Successive Transfer Function Frequency Bins Using the VRM Algorithm	95
C.3	Damping Ratios Computed From Successive Transfer Function Frequency Bins Using the VRM Algorithm	95
C.4	Simulated Frequency Response for Two Serially Linked Second Order Systems	96
C.5	VRM Calculated Natural Frequency in a Noisy Environment	97
C.6	VRM Calculations of Damping Ratio in a Noisy Environment	97
E.1	Example of Peak Flattening Caused by Fixed Order Piecewise Linear Regression	109
E.2	Variable Order Piecewise Polynomial Regression Example	110

GLOSSARY OF ABBREVIATIONS

AR	Auto Regressive – a type of filter or system model whose outputs are a linear combination of a finite number of previous filter outputs.
ARMA	Auto Regressive Moving Average – a system model or filter whose outputs are on a linear combination of the previous filter inputs and outputs.
DFT	Discrete Fourier Transform – any of the implementations of a Fourier transform that is performed on a discrete sequence of time domain information.
DSP	Digital Signal Processor – any processor board specifically designed to implement software for digital signal processing applications such as filtering and transformation of numerical sequences.
ETFE	Empirical Transfer Function Estimation – a widely utilized method of system identification which employs Fourier transformations of both the inputs and outputs of a system to calculate an empirical estimate of the transfer function of the system.
FFT	Fast Fourier Transform – any of the implementations of the DFT, or discrete Fourier transform, that decimates the inputs in time or frequency and then performs bit reversal on the outputs. This process improves computational efficiency from $O(N^2)$ to $O(\log_2 N)$.
FLOPS	Floating Point Operations Per Second – a measure of the speed of a computer processor – the number of floating point multiplications that can be performed by in one second.
FWHM	Full Width Half Maximum – the bandwidth of a power spectrum resonance peak at a height of half its maximum power.
LFIM	Low Frequency Impedance Measurement – a technique for electrochemical system identification developed by Mark Jaworowski and Hobson Lane employing time-domain curve fitting of an output sequence from a system that is excited with a sinusoid input.
LMS	Least Mean Square

LS	Least Squares
MA	Moving Average – a type of filter or system model whose outputs are linear combinations of a finite number of previous inputs to the filter.
MIMO	Multiple Input Multiple Output – a system model or filter that includes any number of input and output signals.
OATF	Optimal Arbitrary Time-Delay Filter – an input shaping algorithm developed by David P. Magee and Dr. Wayne J. Book that relies on a time delay fixed at an optimal value while the filter coefficients can be varied in order to modify the frequency response of the filter.
RALF	Robot Arm, Large and Flexible – a two degree of freedom flexible link device designed as a testbed for verifying and refining vibration control techniques. Developed by Thomas Rowe Wilson and Dr. Wayne J Book.
RISC	Reduced Instruction Set Computing – a computing architecture that implements a limited number of instructions in order to increase the speed at which those instructions can be executed. A highly efficient architecture for mathematical calculation intensive applications such as filtering and transformation and is thus often chosen for digital signal processors.
RLS	Recursive Least Squares –a filtering technique that employs a linear system model whose coefficients are determined by minimizing the square of the error between the system model outputs and the actual observed outputs. Previous model solutions are utilized to recursively solve for successive model coefficients. This filtering technique has proven useful for both system identification and output noise reduction.
SAMII	Small Articulated Manipulator II – a three degree of freedom, rigid link manipulator intended as an end effector for flexible link manipulators. It was developed by David W. Cannon and Dr. Wayne J. Book for use as a testbed for micro/macro manipulator dynamic interactions and control.
SISO	Single Input, Single Output – a system model or filter that includes only one input signal and one output signal.
SNR	Signal to Noise Ratio – a ratio of the power contained in a desired signal to the power in the background noise it is immersed in.
TI	Texas Instruments

- TTFE** Time Varying Transfer Function Estimation – an algorithm for system identification, developed by Tzes and Yurkovich in 1990, similar to ETFE but allows for smoothing in the frequency domain and an RLS filter in the hybrid frequency-time domain.

GLOSSARY OF NOTATION

$f(\cdot)$	Parentheses denote functions of real or complex numbers.
$f[\cdot]$	Brackets denote discrete functions of integers, usually time series.
H_{ij}	Transfer function from input i to output j .
$O(\cdot)$	"On the order of..." or "Increases no greater than..." Unless otherwise specified, the variables m , a , and f within the parenthesis refer to multiplication, addition and transcendental function operations. Subscripts will indicate complex or real number operations. For example, $O(m_c + a_r + f_c)$ may be read as "on the order of four complex multiplications and divisions, plus three real additions and subtractions, and 2 complex transcendental function calls (sinusoids, exponentials, etc.). For the DSP hardware used in this research these values represent minimal time values: $m_r = a_r > 2 \times 10^{-8}$ s, $m_c > 1.2 \times 10^{-7}$ s, $a_c > 4 \times 10^{-8}$ s.
P_{ij}	Auto-correlation power spectrum from input i to input j .
P_{iy}	Cross-correlation power spectrum from input i to output j .
\mathbf{X}	Bold, not italicized, capital, variables represent matrices.
\mathbf{x}	Bold, not italicized, lower-case, variables represent vectors.
x	Italicized, lower-case variables represent scalars.
X	Italicized, upper-case variables are generally used to represent frequency transformations of time-domain scalars, e.g. $X(\omega) = \text{FFT}(x(t))$. However, to retain compatibility with several cited works, N is used to represent a scalar, time-domain integer, usually a buffer length.

SUMMARY

Research into the manipulation of flexible structures has matured to the point that numerous effective means for reducing vibration through active control have been developed and demonstrated. Successful implementation of such algorithms requires knowledge of the dynamics of the controlled device. In addition, systems designed for practical use exhibit time-varying and nonlinear properties that are difficult to predict and model. To eliminate these sources of difficulty, on-line system characterization techniques can be implemented to control the structure in an adaptive manner and further reduce vibration.

The effectiveness of real time system identification and adaptive control of an elastic manipulator was demonstrated on a device named RALF, Robotic Arm, Large and Flexible, a planar, two-degree-of-freedom robot. Vibration information was obtained from lateral-effect photodiodes installed on RALF. Spectrum identification and analysis was performed on a dedicated digital signal processing board located within the controlling MVME bus.

The practical issues examined include choice of sensors, efficiency and accuracy of FFT-based data processing and identification algorithms, and identification response times to changing structure characteristics such as continuous changes in link orientation with respect to gravity. In addition the problem of obtaining vibration data from a structure that is controlled to minimize excitation of such vibration is explored.

CHAPTER I

INTRODUCTION

This chapter motivates research in real-time system identification and adaptive control of flexible structures by pointing out applications and then examines recent theoretical and experimental research in this area. In Chapter II the system identification methods employed in this research are developed. Chapter III describes the hardware and software utilized for implementation of the proposed frequency domain system identification and adaptive control scheme. In addition, issues relevant to the implementation of the optimal arbitrary time-delay filtering technique developed by Magee and Book and its applicability to adaptive control are discussed. Chapter IV provides experimental results from adaptive control experiments on a planar, two degree-of-freedom, robotic device with flexible links named RALF as well as simulations intended to verify the experimental results. Finally, Chapter V provides conclusions regarding the system identification and adaptive control techniques demonstrated. Chapter V will also list contributions of this research, ideas for future work in the area of adaptive input filtering, and explanations of difficulties for future researchers to avoid.

1.1 Adaptive Control Motivation

1.1.1 Manipulator Vibration Reduction Problem

When an engineer designs a robotic device, the rigidity of the joints and links is often given high priority. Rigid robots provide many advantages over flexible manipulators including simplified forward and inverse kinematics formulation, and improved tip positioning accuracy when using simple PI, or proportional-integral, controllers. However, many applications demand performance that exceeds the limits of current materials if confined to rigid designs. For example, the weight of structures deployed in space is limited by the payload capacity of an economical launch vehicle. As a result, elastic structures could provide the optimal solution to this balancing of structure payload with weight restrictions.

In addition, many terrestrial devices, such as the Light Duty Utility Arm, or LDUA, being developed for the Department of Energy, possess significant flexible modes in the manipulator links resulting from design requirements of extreme reach and payload. This device is intended to be deployed in nuclear waste storage tanks through an aperture of only one to four feet in diameter while reaching the sides of a tank forty-five to seventy-five feet in diameter. Even existing equipment, such as power-line maintenance arms, and street sweepers with hydraulically actuated trash vacuuming arms, could benefit from vibration reduction technologies. As a result of these and other applications of elastic robotic manipulators, identification and control of the vibrational modes of these structures has become an area of increasing interest.

1.1.2 Hardware Design Solutions

Many proven hardware design techniques are available to the engineer to reduce endpoint vibration in elastic manipulators. Passive damping of vibration in a robotic device can be accomplished by adding damping material to the structure. However, because the damping material generally cannot contribute to the structural strength of the robot, the payload capacity of the device is necessarily reduced. In addition, passive damping can reduce a structure's residual vibration but cannot eliminate it [2]. Inputs at the resonant frequencies of the passively damped system will still excite significant and possibly destructive vibration.

Another hardware design option is to install vibration absorbers at the resonant frequencies of the links being manipulated. However, this technique requires accurate knowledge of the structure's dynamics. If the true dynamics vary over time or with robot configuration, the robustness of the vibration damping system to these variations becomes critical. Any multi-degree-of-freedom, robotic device, especially those operating in a gravitational field, will experience significant changes in the frequency characteristics of its links. For example, RALF typically experiences a 50% variation in natural frequency over its workspace. Even if a system is precisely modeled and the vibration absorbers are capable of automatically adjusting their frequency characteristics, successful implementation will merely result in two new resonant peaks located above and below each of the original resonance frequencies of the system. These additional peaks must be avoided by any system inputs so as not to induce vibration. As an example, Figure 1.1 shows the frequency response of a second order system with a damping ratio of 5% and a

damped natural frequency of 1 rad/sec coupled with a vibration absorber with the same resonant frequency but a smaller damping ratio of 0.001. It can be seen that the region of large vibration is merely separated and shifted. The maximum magnitude of the response function is not significantly reduced.

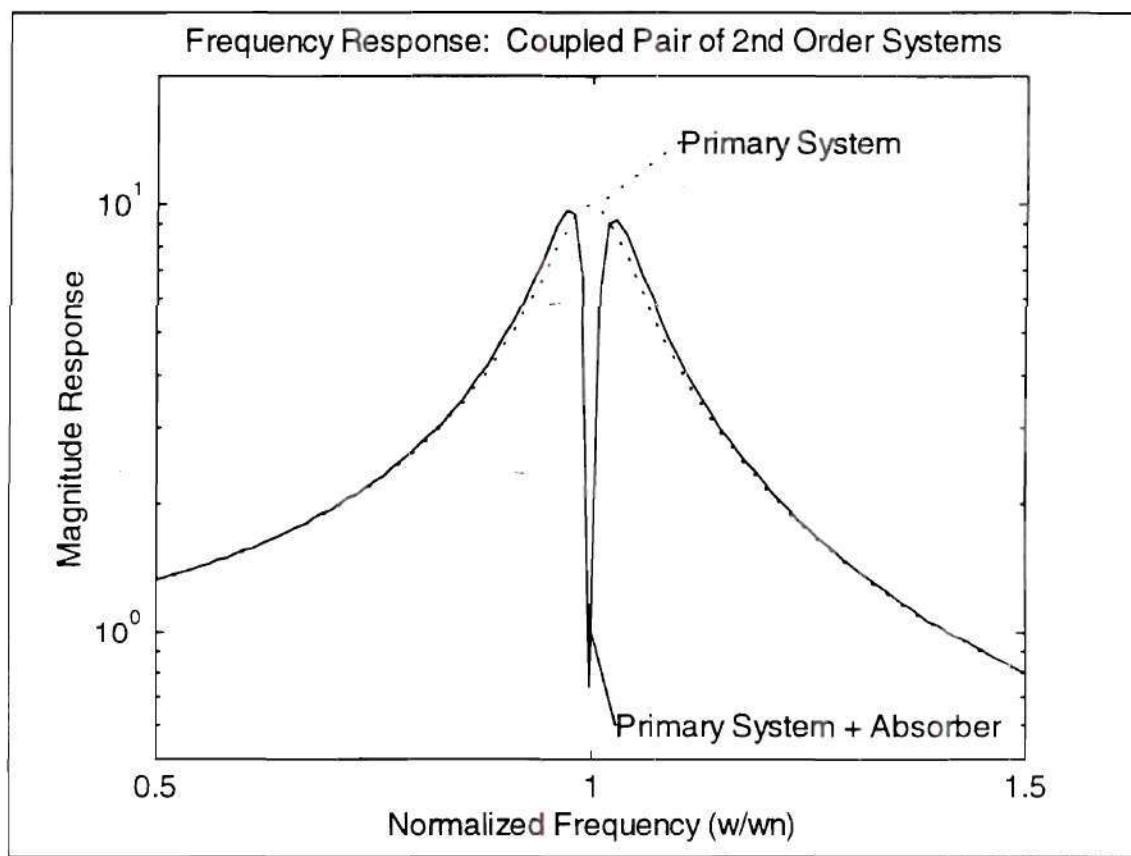


Figure 1.1 Second Order System With a Vibration Absorber Installed

1.1.3 Controller Design Solutions

The most attractive solution to the problem of vibration reduction in elastic robotic devices is to design a controller that either avoids excitation of the manipulator's flexible modes or rapidly nullifies those oscillations whenever they occur. Software solutions have

the additional advantage of being easily retrofitted to existing devices. Vibration reduction can be achieved through a variety of control schemes.

Notch filters can be applied to the inputs of the system at the resonant frequencies of the device to avoid exciting vibration. A similar method, generally referred to as input command shaping, consists of convolving the manipulator input signal with a series of pulses that are selected so as not to excite the natural frequencies of the manipulator [23]. Effectively, input shaping filters attempt to place zeros at or near the known pole locations. The particular input filter utilized for this research was the Optimal Arbitrary Time-Delay filter developed by Magee and Book. A third control method, inertial damping, is accomplished by manipulating a smaller rigid device, or micro-manipulator, at the tip of a flexible link in such a way as to counteract the tip motion with the inertial reaction forces of the micro-manipulator [3][4].

Regardless of the technique employed to reduce manipulator vibration, the natural frequencies and damping characteristics of the system must be available to the designer of the control algorithm or must be measurable states within the controller itself. This necessitates an accurate means of determining the flexible mode parameters of the manipulator. In addition, because robotic devices often have modal parameters that are configuration dependent, a real-time identification scheme must be developed to combat the nonstationarity of a linear approximation of the system.

1.1.4 Adaptive Control Solutions

Adaptive control attempts to overcome the limitations of traditional control schemes when applied to nonstationary systems by identifying the changing parameters of the controlled system in real time and adjusting control gains or input filter parameters accordingly. The volume of literature devoted to the problem of system identification is extensive. This paper will limit discussion to those techniques most suited to real-time application due to their low computational requirements and robustness to sensor noise and other sources of error. A general overview and derivation of the RLS, or Recursive-Least-Squares, technique is provided in Appendix D. RLS is useful not only as a system identification tool in the time domain but is also applicable to the more general problem of noise reduction in time-series data. Such data can include frequency domain information that is evolving in time, or “hybrid time-frequency domain” data [27].

1.2 Prior Work in System Identification for Adaptive Control

Methods for estimation of a linear transfer function from discrete empirical data have been widely studied and employed in both real-time and off-line applications. Research in discrete, linear, transfer function estimation in the frequency domain began with the advent of the discrete Fourier transform that enabled frequency information to be quickly obtained from time series measurements. Even before the advent of the Fourier transform, early least-squares techniques provided a means of estimating discrete linear transfer functions in the time domain. Some examples of time-domain system identification algorithms include the Yule-Walker and Modified Yule-Walker method [18],

and Prony's method [14]. Subsequent refinement has progressively improved the computational efficiency, numerical accuracy, and robustness to noise of transfer function estimation methods.

For on-line system identification and adaptive control, frequency domain techniques are often preferred to time domain techniques for several reasons. Frequency domain algorithms generally require fewer computations, $O(N \log_2(N))$ for the discrete Fourier transform as opposed to $O(N^2)$ for a typical time-domain identification algorithm, where N is the number of frequency bins of information required. The additional computational load required to parameterize the frequency domain data after the Fourier transform has been computed is small, as the poles are readily available as the local maxima of the magnitude spectra. Another advantage of frequency domain techniques is that detailed statistical knowledge about the distribution of noise signals transformed by a discrete Fourier transform is available [8]. As a result, frequency domain identification techniques are generally less sensitive to sensor noise [27]. Also, frequency domain techniques can be used to identify a larger dynamic range of possible transfer functions than equivalent time-domain approaches [7]. Finally, once the slight computational investment of a Fourier transform is paid, frequency information is readily available and can be displayed visually to aid the control designer in verifying control and identification algorithms.

1.2.1 Fundamentals of Frequency Domain System Identification, ETFE

The most basic frequency domain approximation of a discrete transfer function can be found by applying a discrete Fourier transform to a series of the sampled outputs and inputs of a physical system, and then taking the ratio of this frequency domain information at each available frequency bin. This method can be summarized in an input/output relationship as [16]

$$Y_i = H_i X_i \quad i = 0, 1, 2, \dots, N, \quad (1-1)$$

or solving for the unknown transfer function,

$$H_i = \frac{Y_i}{X_i} \quad i = 0, 1, 2, \dots, N, \quad (1-2)$$

where

$$\begin{bmatrix} X_0 \\ \vdots \\ X_N \end{bmatrix} = F \begin{bmatrix} x[0] \\ \vdots \\ x[N] \end{bmatrix}, \quad (1-3)$$

and

$$\begin{bmatrix} Y_0 \\ \vdots \\ Y_N \end{bmatrix} = F \begin{bmatrix} y[0] \\ \vdots \\ y[N] \end{bmatrix}. \quad (1-4)$$

The $x[k]$ and $y[k]$ sequences are the observed, physical system, input and output time series data of length N . The F operator refers to the discrete Fourier transform which can be

implemented by a variety of efficient algorithms including the popular FFT, or Fast Fourier Transform. Tzes and Yurkovich refer to this system identification technique as Empirical Transfer Function Estimation, or ETFE. They assert in published work [29][27] that when this method is utilized, “the variance in the estimation is equal to the signal-to-noise ratio at the frequency under consideration.” Variance reduction of a time-invariant system can be accomplished by averaging the Fourier transforms of nonoverlapping sample ensembles. The variance reduction for such a technique is equal to a factor of $1/n$ where n is the number of averaged ensembles [6]. A significant improvement in the variance of the transfer function produced by this traditional method is gained by averaging multiple Fourier transforms that are performed on consecutive sample sets or ensembles. Using this method the random error in the computed spectrum is reduced by a factor of $1/\sqrt{n}$ where n is the number of averaged, possibly overlapping, sample ensembles [15]. This error estimate of averaged successive spectra is only valid for stationary systems when a large number of consistent sample ensembles can be acquired.

1.2.2 Time-varying Transfer Function Estimation, TTTE

In 1990 Tzes and Yurkovich refined the ETFE approach for application to time-varying systems and renamed it, Time-varying Transfer Function Estimation, or TTTE. This method attempted to reduce the variance of the estimated frequency response while maintaining tracking of a time-varying system [29]. The structure of this time-dependent frequency domain information coined the “hybrid time-frequency domain”[27] by Tzes and Yurkovich is defined by

$$\begin{aligned} \begin{bmatrix} U_0[k] \\ \vdots \\ U_{N-1}[k] \end{bmatrix} &= F \begin{bmatrix} u[k-N+1] \\ \vdots \\ u[k] \end{bmatrix} \\ \begin{bmatrix} Y_0[k] \\ \vdots \\ Y_{N-1}[k] \end{bmatrix} &= F \begin{bmatrix} y[k-N+1] \\ \vdots \\ y[k] \end{bmatrix}. \end{aligned} \quad (1-5)$$

The time dependence of the frequency domain information is now explicitly indicated by the time step index k .

The first substantive improvement over traditional ETFE suggested by Tzes and Yurkovich was to smooth the transfer function in the frequency domain according to

$$H_i[k] = \frac{\sum_{j=(i-\Delta_i) \bmod N}^{(i+\Delta_i) \bmod N} \epsilon_j^i H_j[k]}{\sum_{j=(i-\Delta_i) \bmod N}^{(i+\Delta_i) \bmod N} \epsilon_j^i} \quad (1-6)$$

where Δ_i is the smoothing window width, and ϵ_j^i is the linear weighting factor that determines the degree to which the transfer function value in the i^{th} frequency bin depends upon the element in the j^{th} bin. However, Tzes and Yurkovich did not implement this smoothing function in a real-time control loop due to its high computational cost. In addition, smoothing will often bias the spectrum resulting in damping ratio estimates that are larger than those from an unsmoothed spectrum [21].

The second improvement suggested by Tzes and Yurkovich is more applicable to real-time system identification. A dependence of a given impulse response on the β preceding impulse responses is assumed such that

$$H_i[k] = \chi(H_i[k-1], \dots, H_i[k-\beta_i]) \quad (1-7)$$

This relationship results in the new input/output model

$$Y_i(k) = \sum_{j=1}^{\beta_i} H_i(k-j)U_i(k-j). \quad (1-8)$$

As before, in Equations (1-6) through (1-8), the subscript i enumerates the frequency bin, while the independent variable k is the time step at which the frequency information was acquired, and β_i represents the window width over which a history of transfer functions continues to influence the present response. Notice that the window of past influence, β_i , may be different for each frequency bin in the discrete transfer function.

This formulation is a generalization of the well-known ensemble averaging technique mentioned previously. Equation (1-8) gives the most general expression for the dependence of the present spectrum on previously acquired spectra. However, in practice, the function χ may be chosen such that the frequency response of the system is not calculated at every sample. For example, data could be acquired continuously while recomputing the transfer function estimate only after each l samples have arrived, where l is

an arbitrary positive integer. The batch size, l , should be chosen as small as possible while remaining within the computational limits of the controller. Larger l allows a greater sampling rate for a given computational speed. Smaller l retains more of the variance reduction benefits of TTFE spectrum averaging.

1.2.3 Real-Time Application of TTFE

For real-time application, Tzes and Yurkovich simplify the first part of the TTFE algorithm shown in Equation (1-6) by setting Δ_i to zero for all i . It is then stated that this results in the frequency information merely being windowed according to the ϵ_i^j discrete weighting function [27]. In reality, it can be seen in Equation (1-6) that the weighting factors will actually cancel each other out leaving no effect on the spectrum at all. This is the first typographical error in Tzes & Yurkovich's published work.

Equation (1-6) expresses the most general form of piecewise smoothing of the frequency spectrum. One possible implementation is to employ piecewise least squares polynomial regression on the data. In addition, Equation (1-6) also allows for variable window widths and weighting coefficients that can be implemented as variable window widths and polynomial order in the piecewise polynomial fit. In a simulation discussed in Appendix E this formula was implemented as an automatically variable polynomial order, piecewise curve fit. However, the high computational requirements of solving the poorly scaled matrix equation that results from polynomial regression of power spectrum data near poles made this particular implementation impractical for experimental work.

For the second improvement formulated in Equations (1-6) and (1-7), Tzes and Yurkovich suggest applying N separate, first order, SISO, recursive-least-squares filters to each frequency bin of the estimated transfer function. Appendix D provides a more detailed derivation of the RLS algorithm utilized by Tzes and Yurkovich and extends the method for application to multi-input/multi-output systems. The result is the following pair of recursive relationships for updating transfer function estimates [29]:

$$H_i(k) = H_i(k-1) + \frac{P_i(k-1)U_i(k-1)}{\alpha_i + P_i(k-1)U_i(k-1)^2} [Y_i(k) - U_i(k-1)H_i(k-1)], \quad (1-9)$$

and

$$P_i(k) = \frac{1}{\alpha_i} \left[P_i(k-1) - \frac{P_i(k-1)U_i(k-1)^2}{\alpha_i + P_i(k-1)U_i(k-1)^2} \right]. \quad (1-10)$$

These formulae correct the second typographical error found in Tzes and Yurkovich's work. Implementation of these formulae can be seen to require operations $O(7m_c+3a_c) = O(28m_r+20a_r)$. In addition, considerable care must be taken to maintain the numerical stability of these recursive solutions if more than 100,000 iterations are to be performed [14].

1.2.4 Parameterization of the TTTE

Once a satisfactory complex valued transfer function estimate has been produced it must then be parameterized to be useful for adaptive control. The poles and/or zeros of the

system must be identified. Simple maxima searches over arbitrary ranges of the magnitude spectrum were employed by Tzes and Yurkovich. Determining damping ratio in real time was not attempted. Another technique is to employ a high order polynomial least squares regression of the magnitude spectrum and utilize the analytical derivative of this polynomial to locate the maxima and perform a full width half maximum calculation on the polynomial to determine the damping ratio of the pole. An even more computationally complex solution is to perform a nonlinear curve fit to the data near a peak using a second order system magnitude function to directly determine the damping ratio and natural frequency parameters.

Unfortunately it is this parameterization step that is least often addressed adequately in recent literature on adaptive control. Besides direct maxima searches and full width half maximum techniques, more robust algorithms for spectrum parameterization have been widely available since the 1970's. These include parametric curve fitting, and the method of moments [21]. Rather than relying on the single data point at the peak of a spectrum, all of the points in an arbitrary region around the peak are utilized to calculate a damping ratio and natural frequency. The systems of equations that result from both of these methods involve transcendental functions and must be solved iteratively.

More recently, Lin, Lim, and Liew successfully demonstrated a new modal parameter estimation technique in 1995 called the Variable Residue Method, or VRM [10]. VRM allows identification of closely spaced poles and provides an analytic solution for the damping ratio and natural frequency from complex valued spectrum information contained at only three frequency bins. Again, effort is usually concentrated in regions surrounding

peaks. However, an initial estimate of the pole location is required in order to restrict the algorithm to the region most likely to contain the modal frequency being determined. A detailed description of this method and its merits and limitations is provided in Appendix C.

1.3 Prior Work in Control of Vibration in Mechanical Structures

1.3.1 Flexible Dynamics Control

Because implementations of notch filtering of controller inputs are often noncausal and can result in significant delay times that make teleoperation impractical, input command shaping is the most current area of research and has provided dramatic improvements in system vibration damping. A feed-forward command preshaping technique developed by Singer and Seering successfully improved system performance of the Space Shuttle Remote Manipulator in simulations at Draper Laboratory [24]. This method is more applicable to teleoperation than the notch filter scheme. However, significant delay times often result from utilization of this technique as well. An optimized input shaping technique developed by Magee and Book has been demonstrated to reduce vibration in RALF by 60% while also reducing the delay times to 1/3 of those required by Singer and Seering's technique [12]. The optimized command shaping technique provides the capability of reducing the filter delay times in exchange for increased system vibration amplitude.

Inertial damping techniques have been demonstrated by David Cannon in recent experiments using a device named SAMII attached to the tip of a long, flexible, beam oriented both horizontally and vertically in a gravitational field [4][3]. In both cases tip vibration was significantly reduced and external disturbances were quickly nullified.

1.3.2 Adaptive Control for Vibration Reduction

Utilization of system identification techniques for adaptive control of flexible structures has only recently been demonstrated by researchers such as Tzes and Yurkovich [11]; Milford and Asokanthan [13]; and Yang, Yang, and Kudva [31]. Only in the last few years were these adaptive techniques for minimization of manipulator end-point vibration extended to include multi-degree-of-freedom manipulators, though these testbeds were operated in a horizontal orientation, perpendicular to the force of gravity. Khorrami, Jain, and Tzes demonstrated the TTTE, or time-varying transfer function estimation, algorithm on a manipulator with two flexible links and rotational joints in an adaptive control loop in 1993 [9]. However, all previous experimental work that applied adaptive input shaping or filtering techniques to a two degree-of-freedom robotic device did so for predetermined joint slews rather than continuous operation. In addition the nonlinear effects of a gravitational field were always eliminated by hardware and experiment design. The work of this thesis distinguishes itself by applying adaptive input filtering techniques to a two degree-of-freedom device operating in a gravitational field under continuous control and experiencing configuration-dependent variations in modal characteristics.

The work by Khorrami, et al., in 1995, most closely resembles the emphasis of this thesis and is discussed here in more detail. The Singer and Seering method of input preshaping was utilized for inputs to a nonlinear controller with a standard PD inner loop. The nonlinear controller compensated for the first order Coriolis and centripetal terms in the rigid body dynamics of the two link testbed. For the input shaping algorithm, the amplitude of the first system input pulse was computed based on a combination of the commanded input and the predetermined damping of the system. No on-line identification of the damping ratio was implemented as this parameter changed little with the addition of a payload. The delay time and amplitude of the second impulse were determined by the payload-dependent first natural frequency of the device. This quantity was calculated during the delay between application of the first pulse and the second by computing a 2048 FFT on a limited sample set of the system response padded with zeros. The sample duration was inadequate during the interim between application of the first two pulses to resolve the first fundamental frequency at 0.8 Hz, but sufficient for computing the second mode at 8.8 Hz. An estimate of the first mode was then calculated based upon prior knowledge of the variation of these modes with changing payload.

In this way, both the delay time and the filter coefficients of the Singer and Seering filter were adapted in real time in response to a discrete change in payload. This method reduced maximum tip acceleration by 41% after a 60° slew of both joints for a discrete fundamental resonant frequency change greater than 80% [9]. Similar reductions in residual tip deflection oscillation amplitude were reported. However, experiments in continuous automated and/or teleoperated control were not performed by Khorrami et al.

and the proposed scheme does not account for the overlapping pulse trains that would result from such an implementation.

The Optimal Arbitrary Time-delay Filtering method developed by Magee and Book has been implemented in real-time continuous teleoperation experiments and was therefore chosen as the control method for experiments presented in this thesis. For the experiments performed for this research, continuous, automated trajectories, the time-delay values of the filter remained fixed during each experiment to ensure applicability to teleoperation tasks. Because of the more ambitious goals of this research to apply the adaptive input filtering techniques to the practical situation of continuous control in a gravitational field, improvements in tip vibration reduction beyond those experienced by Khorrammi et al. are not expected.

CHAPTER II

FREQUENCY DOMAIN SYSTEM IDENTIFICATION

2.1 Introduction

This chapter first poses the problem of identifying a linear system from a history of its inputs and outputs. Section 2.2 then provides a justification for the transfer function representation of the testbed utilized for the experimental work of this research. Finally, Section 2.3 presents several traditional system identification algorithms and proposes modifications for the work of this research.

2.1.1 Problem Statement

System identification is the attempt to determine the general linear and/or nonlinear effects of inputs on a system from empirical data. The system might be physical, such as a mechanical device or an electrical circuit, or might represent input-output relationships that relate less directly to physical phenomena such as economic and social group dynamics. In controls engineering this is often formulated as the impulse response function of a linear or nonlinear transfer function representing a physical system. The goal of system identification is to predict future outputs of a system from a history of past inputs and outputs of the system. This then provides the means to design an effective controller to drive the system to the desired states. Extensive literature is widely available on the subject

of system identification. This research will concentrate on those techniques most applicable for real-time control of robotic devices.

System identification is subdivided into several categories. One division is between parametric and nonparametric methods. As its name suggests, a parametric method attempts to determine the value of a finite number of numerical values or parameters, possibly related to physical characteristics of the system, that accurately determine the response of the system to inputs. Nonparametric methods allow the more general modeling of system characteristics as a difference equation with an infinite number of terms. Often further analysis is required to glean useful information from this data.

In addition, system identification schemes are often categorized according to the system type they are designed to identify: linear or nonlinear, and time variant or invariant. A linear system is one whose input-output relationship obeys the superposition principle. A nonlinear system is one for which the system parameters depend upon the configuration of the device. In other words, the differential equation relating inputs to outputs contains nonlinear terms. If the nonlinearities of a system do not dominate its response, such a system is often approximated by a linear model due to the decreased complexity of the mathematics involved and the extensive body of previous work on the subject. In addition to nonlinearities, the time varying nature of some systems can be overcome by utilizing adaptive versions of traditional system identification algorithms [20]. Thus, in order to implement real-time control of a nonlinear system it is often necessary to implement adaptive control which itself utilizes adaptive system identification.

The robotic device being considered for identification, RALF, exhibits nonlinear and time-varying system characteristics. Linearization about an operating point can be accomplished to simplify identification and control design. However, the time-varying nature of RALF cannot be avoided. In addition, in order to utilize information from a system identification algorithm to update the coefficients of an input shaping filter, system parameters must be made available to the controller. Thus, the system identification schemes considered in this research provide parametric models of linear, time-varying systems.

2.2 Dynamic Model of Testbed

The first step in system identification is to determine a system model. Because modeling of the dynamics of RALF is not the core research area of this thesis, a simplified, assumed modes, modal damping model was employed. A general equation of motion for the multi-degree of freedom time-variant flexible structure such as RALF is given by [22]

$$\mathbf{M}(t)\ddot{\mathbf{q}}(t) + \mathbf{D}(t)\dot{\mathbf{q}}(t) + \mathbf{K}(t)\mathbf{q}(t) = \mathbf{E}(t)\mathbf{f}(t), \quad (2-1)$$

where $\mathbf{q}(t)$ represents a vector of the general modal coordinates and \mathbf{M} , \mathbf{D} , \mathbf{K} , and \mathbf{E} represent the mass, damping, stiffness, and input influence matrices of the structure being considered. Coriolis, and centrifugal effects are assumed to be lumped into the damping matrix \mathbf{D} . Assumption of time-invariance produces

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{D}\dot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) = \mathbf{E}\mathbf{f}(t). \quad (2-2)$$

2.2.1 State Space Representation

State space representations are generally desired for controller implementation. One can combine the modal coordinates in a vector \mathbf{q} . This vector and its first derivative are then combined to produce the state vector \mathbf{x} . The available inputs to $\mathbf{f}(t)$ can be mapped by the input matrix \mathbf{B} from an input vector \mathbf{u} . Lastly, the measurable states are mapped by the output matrix \mathbf{C} to produce

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} \end{aligned} \quad (2-3)$$

An alternative discrete representation is given by

$$\begin{aligned} \mathbf{x}[k+1] &= \tilde{\mathbf{A}}\mathbf{x}[k] + \tilde{\mathbf{B}}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{Cx}[k] \end{aligned}, \quad (2-4)$$

where k is the discrete time index.

2.2.2 Transfer Function Representation

Accurate modeling of all of the dynamics of the system represented in the state-space model are not necessarily required in order to implement effective control. The system pole locations are the most critical information for minimization of vibration. In

particular the arbitrary time-delay filtering method developed by Book and Magee only requires the modal frequencies and damping ratios of system resonances to successfully filter out those frequency components. Such a requirement lends itself to the transfer function approach, where any linear system can be represented by a combination of frequency response functions. The denominators of these transfer functions determine the pole locations and thus the modal characteristics of the system. Any state space model can be converted to a transfer function representation. The transfer function of the state space model shown in Equations (2-3) and (2-4) can be found to be [26]

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}, \quad (2-5)$$

where $\mathbf{H}(s)$ is the transfer function obtained from the state-space system model, and s is the complex frequency variable. Thus, it can be easily seen that the determinant of the $(s\mathbf{I} - \mathbf{A})$ term in the transfer function given by Equation (2-5) determines its poles. It is also obvious that this determinant will result in a polynomial in s , the complex frequency variable. This polynomial in s can then be represented as a product of second order polynomials in s such that

$$\det(s\mathbf{I} - \mathbf{A}) = \prod_{r=1}^N (s^2 + 2\zeta_r \omega_r s + \omega_r^2), \quad (2-6)$$

where ζ_r , and ω_r represent the damping ratio and modal frequency of the r^{th} vibrational mode of the system.

Further examination of Equation (2-6) shows that the individual poles dominate the frequency response in the regions around their damped natural frequencies. As long as these poles are not closely spaced, the transfer function can be assumed to consist of a single pole within that small region near a pole. As a result of these simplifications, the problem of system identification of a complex structure such as RALF has been reduced to the more manageable task of identifying and tracking the r^{th} pole or identifying the parameters of,

$$H_r(s) = \frac{a_r}{(s^2 + 2\zeta_r\omega_r s + \omega_r^2)} + h_r \quad (2-7)$$

where a_r is a scaling factor that depends upon the nature of the transfer function being identified, but not on frequency, and h_r contains the residual response from other poles in the region, or from sensor noise.

2.3 System Identification Algorithms

Several system identification algorithms were considered for on-line adaptive control. Those most related to the final identification scheme utilized in this research are briefly discussed here.

2.3.1 Recursive Least Squares

Recursive least squares (RLS) is a technique for fitting a polynomial of arbitrary degree to a continuously updated stream of data in the time domain. It is similar to standard least squares polynomial regression, except that it improves efficiency by not recalculating a fit for all previous data points, but rather only updates the new fit based upon the latest data point and the previous fit statistics. However, the relationship that is fitted to the data must not only be linear in the parameters being fitted, as in polynomial regression, but the relationships determined by those parameters must be linear as well. A typical example is an input-output difference equation where future results are simply linear combinations of previous inputs and outputs. Kalman filtering evolved from this technique and provides a method for optimally filtering noise from an input signal and also provides a means of identifying time-varying systems.

In addition the RLS algorithm has proved useful for reduction of error or variance in the frequency domain. It provides a means for smoothing data within a single frequency spectrum. In the hybrid time-frequency domain, the RLS algorithm has proven useful at reducing the variance of time-varying spectral information by operating on individual frequency bins separately through time. This implementation is a two data point, scalar version of the general RLS matrix operator on an arbitrary, finite ensemble of data points. This RLS implementation is the basis of the time-varying transfer function estimation, TTFE, procedure developed and implemented by Tzes and Yorkovich which was mentioned in Chapter I. As a result of the RLS algorithm's wide applicability to both time-domain and frequency domain data, a complete derivation is provided in Appendix D. The

TTFE algorithm is discussed again briefly in the next section before presenting modifications to the algorithm that were employed for this research.

2.3.2 TTFE

Time varying transfer function estimation, or TTFE, was derived from empirical transfer function estimation, or ETFE, by Tzes and Yurkovich in 1990 as discussed previously in Chapter I. As its name implies, TTFE did not propose a new method for determining the parameters of a linear system, but rather provided a new method for constructing a transfer function from observed data. A simple maximum search was performed on predetermined ranges of the spectrum by Tzes and Yurkovich in experimental work in order to determine the modal frequency from the spectrum. The damping ratio was assumed to be constant. These same techniques were utilized in the work discussed for this research. However, frequency domain smoothing was implemented in the form of a piecewise linear regression smoother which is discussed in Section 2.4.1. In addition, damping ratio determination was accomplished off-line using the Full Width Half Maximum approach discussed in the following section.

2.3.3 Full Width Half Maximum, FWHM

The Full Width Half Maximum, or FWHM, technique is a traditional and widely utilized method for empirically determining the damping ratio of a system resonance from a transfer function estimate near a pole. Its accuracy is limited by the fact that it employs information from only two frequency bins within a discrete transfer function estimate. It's

precision is limited by the resolution of the transfer function estimate. In addition, the method is only applicable to lightly damped systems, $\zeta < 0.1$, due to neglect of higher order ζ terms in the derivation of FWHM relationship,

$$\zeta = \frac{\Delta\omega}{2\omega_n}, \quad (2-8)$$

where $\Delta\omega$ is the measured half-power bandwidth, and ω_n is the observed undamped natural frequency, and ζ is the damping ratio or fraction of critical damping of the transfer function pole. A complete derivation of the relationship in Equation (2-8) is provided in Appendix B. As a result of the accuracy and precision limitations mentioned previously, the FWHM technique was only utilized for off-line analysis of spectra computed during various portions of an experimental trajectory.

2.4 Modifications to TTFE

2.4.1 Frequency Domain Smoothing

Because simple global maximum searches were employed to determine the resonant frequencies of poles, it was deemed necessary to implement the frequency domain smoothing not implemented by Tzes and Yurkovich. Automatic order adjusting piecewise regression was first considered for this task but was later rejected due to its high computational cost. Appendix E provides simulation data from such an adaptive smoother.

Instead, a fixed width, fixed order, piecewise regression was utilized. Such a smoother can be reduced to a linear filter of length equal to the window width through off-line calculation of the Vandermonde matrices and off-line matrix inversion. Refer to the Mitra and Kaiser text for more details on this technique [14].

2.4.2 Hybrid Time-Frequency Domain Smoothing

The second improvement suggested by Newland as well as Tzes and Yurkovich is the averaging of each frequency bin of the current spectra with those of previously acquired spectra. In order to retain the computational benefits of frequency domain system identification over time-domain RLS system identification techniques, it is important not to employ computationally complex algorithms in the frequency domain. The scalar, first order, RLS algorithm employed in the hybrid time-frequency domain by Tzes and Yurkovich, provides a tremendous computational advantage over a high order RLS time-domain system estimate. A scalar RLS algorithm requires computations $O(m_c \cdot p^2)$, where p is the order of the filter. A large order filter, equivalent to an FFT data buffer length, is required for time domain identification with a wide dynamic range. However, the number of operations required for a scalar RLS implementation is only $O(10m_c + 2a_c + 2a_r) = O(40m_r + 26a_r)$ for each frequency bin of required information, and can be performed selectively in the frequency region of interest.

In order to reduce the computations and memory required for such averaging, smoothing in the hybrid time-frequency domain was accomplished using exponential averaging rather than the more complex recursive least squares method suggested by Tzes

and Yurkovich. An exponential moving average in the hybrid time-frequency domain is easily implemented in-place by using the following recursive relationship,

$$H_i[k] \leftarrow (1 - \alpha)H_i[k] + \alpha H_i[k-l] \quad (2-9)$$

where $H_i[k]$ represents the latest complex valued empirical transfer function estimate at frequency bin i and time step k , while α is a forgetting factor between zero and one. Again, l represents the batch size limited by the computational constraints of the computer hardware available. Typical values for α utilized for the experimental work of this thesis were between 0.05 and 0.2, while the batch size ranged between 20 and 50 data points for a 200 Hz sampling rate.

A brief comparison of the computational and storage requirements of RLS and exponential averaging is provided here to justify the selection of exponential averaging over RLS. As mentioned previously, the scalar RLS algorithm requires $O(10m_c + 2a_c + 2a_r) = O(40m_r + 26a_r)$ operations for each frequency bin. In comparison the exponential averaging method requires $O(4m_c) = O(16m_r + 8a_r)$ operations for each complex frequency response value. Finally, the scalar RLS algorithm utilized by Tzes and Yurkovich requires storage of a covariance value for each frequency bin computed in addition to storage of the current spectrum bin being averaged. The exponential averaging method requires no additional storage beyond that required for the frequency response and can be accomplished in-place.

2.4.3 Multiple Input, Multiple Output Modification

Previously discussed methods for estimating the transfer function of a single input, single output, or SISO system such as the ETFE or TTFE algorithms developed and demonstrated by Tzes and Yorkovich are not adequate for articulated manipulators with multiple degrees of freedom. One possible solution is to apply the multi-input/multi-output RLS algorithm provided in Appendix D to the vector of input and output frequency domain information. This provides continuously updated estimates of the transfer functions between all of the inputs and outputs. However, because the matrix RLS algorithm is considerably more complex than the scalar solution, a more efficient solution is sought.

One possible solution is to derive analytic relationships determining the various transfer functions from the frequency domain input and output information prior to applying a scalar RLS filter to each of these estimates. A transfer function block diagram of a two input, two output system is provided in Figure 2.1. The approaches discussed here can be extended to apply to greater numbers of inputs and outputs.

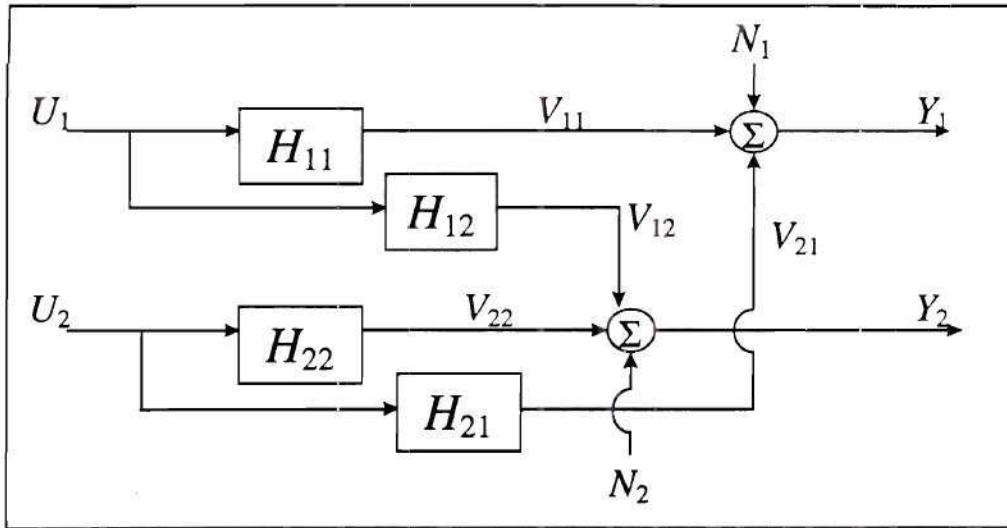


Figure 2.1 Two Input, Two Output, Transfer Function Model

U_1 and U_2 represent the frequency transform of the two inputs $u_1(t)$ and $u_2(t)$. Y_1 and Y_2 represent the frequency spectra of the two measured outputs $y_1(t)$ and $y_2(t)$. N_1 and N_2 represent the frequency components of the two unknown noise signals added to the true outputs of the system to produce the measured outputs. The V signals represent the uncorrupted, true outputs of the system. The H_{ij} blocks show all the possible combinations of linear relationships between the two inputs and the two outputs that might be present in a given system.

Interpretation of the block diagram produces

$$Y_1 = V_{11} + V_{12} + N_1 = H_{11}X_1 + H_{21}X_2 + N_1 \quad (2-10)$$

$$Y_2 = V_{21} + V_{22} + N_2 = H_{12}X_1 + H_{22}X_2 + N_2 \quad (2-11)$$

or, in matrix form,

$$\mathbf{Y} = \mathbf{V} + \mathbf{N} = \mathbf{H}\mathbf{X} + \mathbf{N}. \quad (2-12)$$

Thus we have only two equations and six unknowns. However, additional equations can be obtained by taking advantage of the fact that the input and output spectra are complex numbers and multiplying Equations (2-10) and (2-11) by the complex conjugates of the inputs to obtain

$$U_1^* Y_1 = H_{11} U_1^* U_1 + H_{21} U_1^* U_2 + U_1^* N_1 \quad (2-13)$$

$$U_2^* Y_1 = H_{11} U_2^* U_1 + H_{21} U_2^* U_2 + U_2^* N_1 \quad (2-14)$$

$$U_1^* Y_2 = H_{12} U_1^* U_1 + H_{22} U_1^* U_2 + U_1^* N_2 \quad (2-15)$$

$$U_2^* Y_2 = H_{12} U_2^* U_1 + H_{22} U_2^* U_2 + U_2^* N_2 \quad (2-16)$$

In addition, by taking the expected value of these equations the noise terms will approach zero, assuming that the noise signal is uncorrelated with either of the inputs and has zero mean. This is true for any random noise. In addition, the remaining terms of the equation can be rewritten in terms of power spectra. By converting to power spectrum notation, the importance of taking the expected value before solving for the transfer function estimates is emphasized. While the expectation of a sum is the sum of the expectations, the same is not true of the expectation of a product or quotient of random variables. The new equations are

$$P_{1y1} = H_{11} P_{11} + H_{21} P_{12} \quad (2-17)$$

$$P_{2y1} = H_{11}P_{21} + H_{21}P_{22} \quad (2-18)$$

$$P_{1y2} = H_{12}P_{11} + H_{22}P_{12} \quad (2-19)$$

$$P_{2y2} = H_{12}P_{21} + H_{22}P_{22}. \quad (2-20)$$

Nonetheless, our multiplication by the complex conjugates of the inputs has produced four equations that remain redundant. However, an identity of complex numbers will resolve this problem. After solving for the four unknown transfer functions, and using the commutative property of multiplication of complex numbers which gives $P_{12}=P_{21}$ and $P_{12}P_{21}=|P_{12}|^2$ we obtain,

$$H_{11} = \frac{P_{1y1}P_{22} - P_{12}P_{2y1}}{P_{11}P_{22} - |P_{12}|^2} \quad (2-21)$$

$$H_{12} = \frac{P_{1y2}P_{22} - P_{12}P_{2y2}}{P_{11}P_{22} - |P_{12}|^2} \quad (2-22)$$

$$H_{21} = \frac{P_{2y1}P_{11} - P_{12}P_{1y1}}{P_{11}P_{22} - |P_{12}|^2} \quad (2-23)$$

$$H_{22} = \frac{P_{2y2}P_{11} - P_{12}P_{1y2}}{P_{11}P_{22} - |P_{12}|^2} \quad (2-24)$$

It is important to note that the expectation value was taken on the individual power spectra and not on the empirical transfer function solution. Thus, if a time averaged transfer function estimate is desired, the averaging technique, e.g. an exponential smoother or RLS, must be performed on the individual power spectra before solving for the transfer function estimate. This means that the averaging algorithm will have to be performed 8 times, increasing its proportion of the computational requirements of the entire transfer

function estimation algorithm and justifying the choice of a simple exponential smoother over RLS for the sake of computational efficiency.

If it can be determined that the inputs are uncorrelated with each other such that

$U_1^* U_2 = U_2^* U_1 = 0$, then the Equations (2-21) through (2-24) further simplify to,

$$H_{11} = \frac{U_1^* Y_1}{U_1^* U_1} \quad (2-25)$$

$$H_{12} = \frac{U_1^* Y_2}{U_1^* U_1} \quad (2-26)$$

$$H_{21} = \frac{U_2^* Y_1}{U_2^* U_2} \quad (2-27)$$

$$H_{22} = \frac{U_2^* Y_2}{U_2^* U_2} \quad (2-28)$$

This simplification was not possible for the experimental work on RALF discussed in Chapter IV because the correlation coefficient was determined to range between 0.4 and 0.9, for the tip and joint trajectories utilized, approaching perfect correlation in some cases.

The question remains as to whether it is necessary to empirically measure and calculate all of the transfer functions shown in Figure 2.1. First, it must be determined if the computations required for such an analysis are prohibitive, or even comparable to the computations required to obtain the Fourier transforms of the input and output data in the first place. It is well understood that a fast Fourier transform of a time sequence requires a number of operations given by

$$\Omega_{FFT} \propto \frac{N}{2} \log_2 \left(\frac{N}{2} \right), \quad (2-29)$$

where N is the number of time domain data points being transformed which is double the number of complex numbers produced by the FFT in frequency bins.

The operations required for transfer function estimation by solving any one of Equations (2-21) through (2-24) for the entire frequency spectrum was determined to be

$$\Omega_{ETFE} \approx \frac{N}{2} (9m_c + 1a_c + 1d_c + 2m_r + 2a_r) = \frac{N}{2} (46m_r + 26a_r + 2d_r). \quad (2-30)$$

It can be seen that this value will almost certainly exceed the requirements of the FFT for most practical values for N , the size of the time-domain data buffer. Thus, it would be computationally expensive to compute all four transfer functions present in a dual input-output system. For the uncorrelated inputs case, the number of operations reduces to

$$\Omega_{ETFE}^{uncorr.} \approx \frac{N}{2} (2m_c + 1d_c) = \frac{N}{2} (16m + 6a + 2d). \quad (2-31)$$

2.4.4 Summary of Modifications

The proposed modifications to TTFE reduce the computations and memory required when applied to SISO systems. This is made possible by the reduction of hybrid

time-frequency domain smoothing computations while implementing frequency domain smoothing to compensate for any loss of variance reduction caused by this reduction. This technique is justified by Gardner who proved the equivalence of frequency domain smoothing and hybrid time-frequency averaging of continuous time-dependent Fourier transforms of infinite length [5]. In addition Gardner also asserts the approximate validity of this equivalence for discrete, finite length, Fourier transforms of a time varying system, making it applicable this research. Smoothing in the time-frequency domain reduces time resolution. Smoothing in the frequency domain degrades frequency resolution. Thus a proper balancing of the two considerations is required to maximize the variance reduction of the computed spectra while minimizing the lag time between system changes and the computed spectrum response. This balancing was accomplished by trial and error for the experiments discussed in Chapter IV. In addition, the TTFE computation reductions proposed allow the application of the algorithm to dual input-output systems in on-line applications when compared to a direct implementation of TTFE. Without these modifications, the update rate of the TTFE identification scheme would be drastically reduced, hindering the performance of any adaptive control system.

CHAPTER III

HARDWARE AND IMPLEMENTATION

This chapter provides a brief description of the robotic device, RALF, utilized for the research of this thesis and depicted in Figure 3.1. Section 3.1 limits discussion to the mechanical hardware of RALF. Additional sections of this chapter describe the computational hardware and software utilized for control of RALF as well as the particulars of OATF implementation.

3.1 RALF, Robot Arm, Large and Flexible

Envisioned as a testbed for demonstration of elastic structure control technologies, RALF, consists of two ten foot aluminum links connected by rotational joints to each other and the base. Static deflection of the links by several millimeters is typical throughout RALF's semicircular workspace. The links are both actuated by hydraulic cylinders. A parallel bar linkage is utilized to actuate the second link from a hydraulic cylinder attached to the base.



Figure 3.1 RALF, Robot Arm, Large and Flexible

3.1.1 Sensors

The extension of the hydraulic actuators on RALF was measured through the use of Tempsonics linear transducers, installed parallel to the stroke of each hydraulic cylinder. Calculation of joint angles was then possible through the use of the law of cosines applied to the triangles formed by the pins of the actuators and the links. After amplification and sampling the Tempsonics transducers provided a resolution of 5×10^{-3} in. over a range of 20 in. with standard deviations of 5.4×10^{-3} in. and 4.6×10^{-3} in. for joint 1 and joint 2 respectively. This indicates binary noise due to the 12 bit A/D resolution limit. When these extension measurements were converted to the joint angles, an average resolution of 3.5×10^{-4} rad. and an average range of 1.38 rad. were possible in a major portion of the workspace, though this resolution was severely degraded near singularities, e.g. the entire upper limit of RALF's workspace.

For measurement of the link deflections several sensors were available, strain gages at the base of each link, accelerometers installed at the end-effector, and deflection gages installed on each link. Only deflection gage data was utilized for experimental work in this thesis because of the measurements' direct relationship to the quantities of interest for performing tip positioning tasks. The deflection gage systems consisted of an infrared, lateral-effect photodiode installed at the end of each link, and an infrared digital camera at the base of each link. The resulting measurements provide an estimate of the link deflection parallel to the plane of motion and perpendicular to orientation of the base of each link. After an amplification and anti-aliasing filter was installed the deflection gages provided resolutions of 2.8×10^{-4} in. and 2.7×10^{-4} in. and ranges of 1.15 in and 1.11 in. for

each link respectively. Background noise and link vibration at home position produced standard deviations of 2.1×10^{-3} in. and 3.8×10^{-4} in. for the respective link deflection measurements. In the frequency domain, sensor data exhibited minimal high frequency roll off and a bandwidth greater than 2 kHz.

3.1.2 Actuators

Two Atlas hydraulic cylinders actuate each of the joints of RALF. The first actuator exerts a moment on the first link directly. A parallel bar mechanism transmits the force exerted by the second actuator to the second link, while locating the bulk of its mass near the base. Pressure to the cylinder pistons is regulated by a feedback loop controlling the valves. Because of the light weight construction of RALF and the 1000 psi operating pressure of the hydraulics, the time delay between a voltage applied to the valve controller and steady state cylinder velocity is less than 20 ms. The end result is that the cylinder velocity can be controlled with a very small time constant, on the order of the sampling time of the controller implementations utilized for this research. The forward and inverse kinematic relationship between tip position and joint rotation is provided in Appendix A.

3.2 Computation and Control

This section provides a description of the computation hardware and software utilized for control of RALF through interaction with the sensors and actuators described in the previous section. The computational hardware and its performance capabilities is listed first in Section 3.2.1. A description of the techniques and structure of the software

utilized for this research is provided in Section 3.2.2, while Section 3.2.3 discusses the details of communication between these various software modules.

3.2.1 Hardware

Digital control of RALF is implemented on a Motorola MVME177 CPU board with a 50 MHz MC68060 processor within a Motorola MVME 946 chassis. An additional Motorola MC68040 processor board, also installed in the MVME chassis, was utilized for preliminary testing of control and system identification algorithms. Both processors utilized the VxWorks, real-time, multitasking, operating system. Data input is obtained via two 12 bit, 16 channel, 200 kHz Acromag AVME 9325-5 analog to digital conversion boards within the same MVME chassis. Hydraulic valves are actuated using 2 Acromag AVME 947X Digital to analog conversion boards.

In addition to the core control hardware, a Spectrum DBV-44 digital signal processing board is also located within the VME bus chassis. The RISC architecture of the single Texas Instruments TMS320-C40 chip located in one of the four available TIM-40 sockets on the spectrum is optimized for computation of multiplications and additions. The chip is capable of 275 MOPS (integer operations), 50 MFLOPS (floating point operations), or 25 MIPS (instructions). For data and code storage, 384 Kbytes of RAM were installed in addition to the 2 Kbytes of on-chip RAM. Because of its RISC architecture the TMS320-C40 is ideally suited for filtering and FFT computation. Though the board is not directly connected to the A/D and D/A boards it can be accessed via the controlling CPU across the MVME bus. Routines supplied with the hardware were utilized for passing

large blocks of integers from memory on the CPU board directly to memory on the DSP board without the use of the high speed parallel ports on the DBV-44 board.

3.2.2 Software

The hardware architecture described in the previous section lended itself well to the division of computation loads between the CPU and DSP boards depicted in Figure 3.2. The Optimal Arbitrary Time Delay filtering developed by Magee and Book [12], and joint control were accomplished on the Motorola CPU board, while the system identification functions of transforming data into the frequency domain, computing and smoothing the empirical transfer function estimates, and gleaned pole parameters from the empirical transfer functions, were all performed on the DSP board.

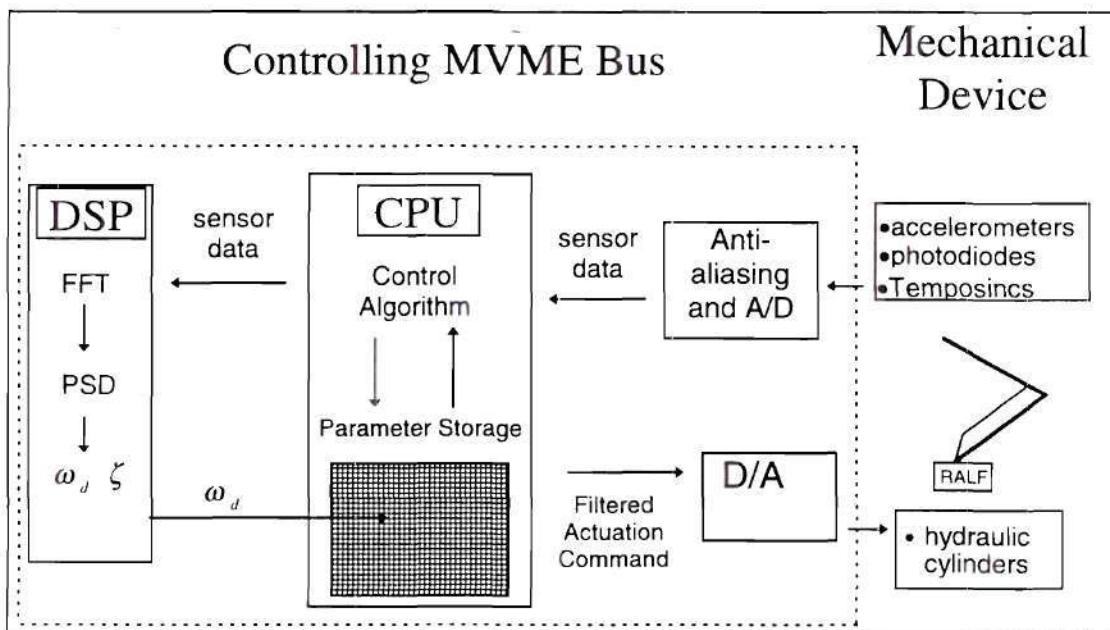


Figure 3.2 Software Module Distribution and Interaction

In order to ensure the usability and modularity of code developed for this research, the majority of the algorithms and data storage performed on the CPU board were encapsulated in C++ objects. In addition, it was possible to package in a single object all of the complex tasks and data required to initialize and communicate with the DSP board in real time. The task of interacting with the A/D and D/A boards was similarly simplified previously by David Cannon in the form of Sensor and Actuator objects. A diagram of the objects created for this research and their relationships to one another is summarized in Figure 3.3.

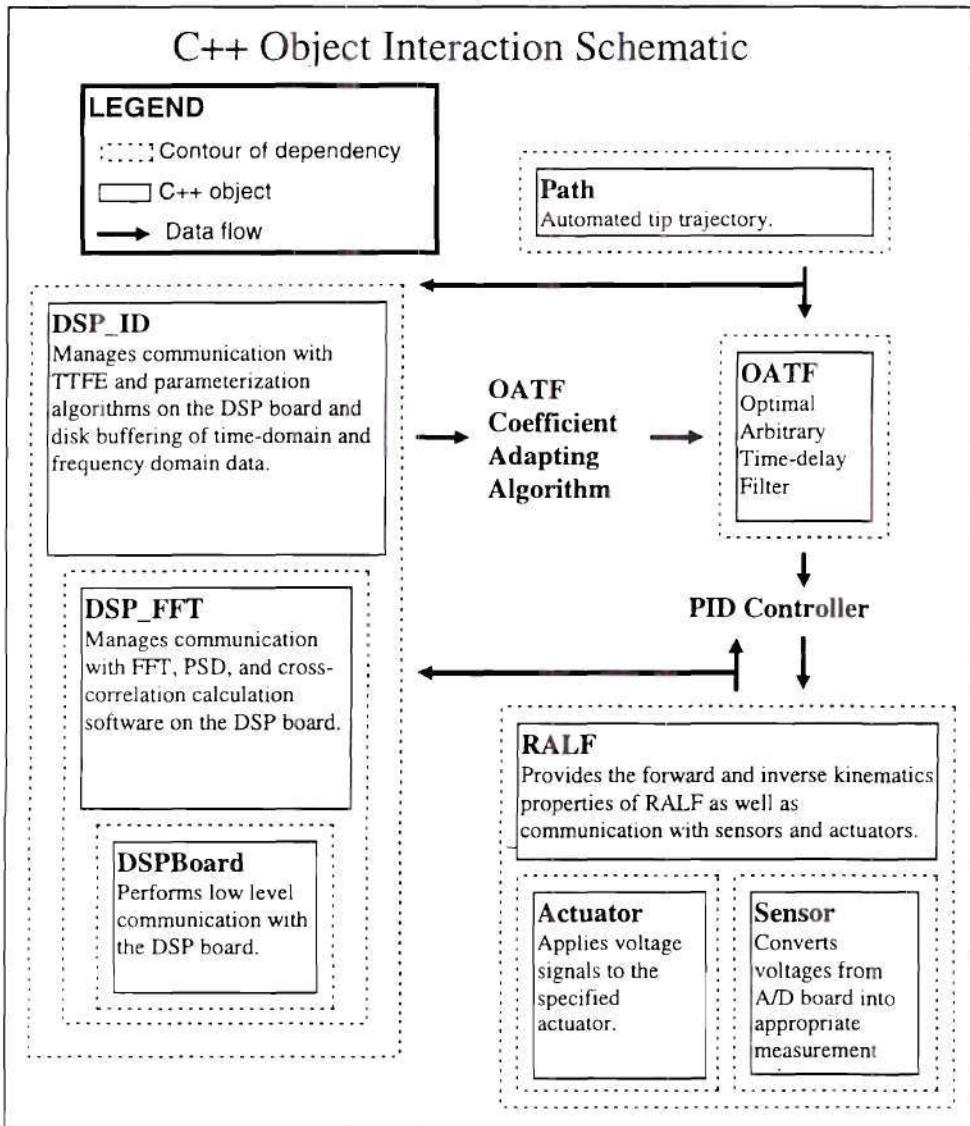


Figure 3.3 C++ Class Hierarchy and Interaction

No C++ compiler was available for the TMS 320-C40 DSP chip. Even if it were possible to compile C++ code for the DSP board, the highly procedural nature of code usually implemented on RISC processors would not lend itself well to segmentation into C++ classes. In fact, greater efficiency is obtained if compiled C code can be avoided all together. This can be accomplished with minimal knowledge of assembly language

programming if the desired functions can be found among the vast library of available assembly language code available for the TMS320-C40 chip. For this research, only the core FFT algorithm was implemented utilizing public domain assembly language code optimized for the TMS320-C40 chip. All other identification and parameterization code was written in C and compiled using the Texas Instruments optimizing C compiler.

A brief quantification of the performance advantages experienced when utilizing assembly language routines on the Spectrum DBV-44 DSP board is provided here. The first test runs were performed using traditional FFT C code taken from a public domain scientific computing package, *Numerical Recipes in C* [19]. For the MC68040 CPU board, the C code was compiled using a standard GNU C compiler. For the DSP board the code was compiled with the Texas Instruments optimizing C compiler set to level 2 optimization. A third test run on the DSP board was performed using assembly language code incorporated into skeleton C code compiled using the TI compiler set at level 1 optimization as required by the register utilization of the assembly language code. It can be seen from Figure 3.4 that the computation time was cut in half by incorporating the assembly language code rather than relying on compiled C code alone for computation of an FFT on the DSP board. The factor of ten that might be expected was not possible in part due to the high degree of optimization already accomplished with the TI compiler. Figure 3.4 charts the speed with which the various processors and coding combinations were able to compute FFTs of length 1024 (2^{10}) to 16384 (2^{14}). It can be seen from this bar chart that the TMS320-C40 DSP chip can compute an FFT using C code in approximately a tenth of

the time required on the general purpose MC 68040 chip running identical code. Likewise, the assembly language routine was more than twice as fast as the C implementation.

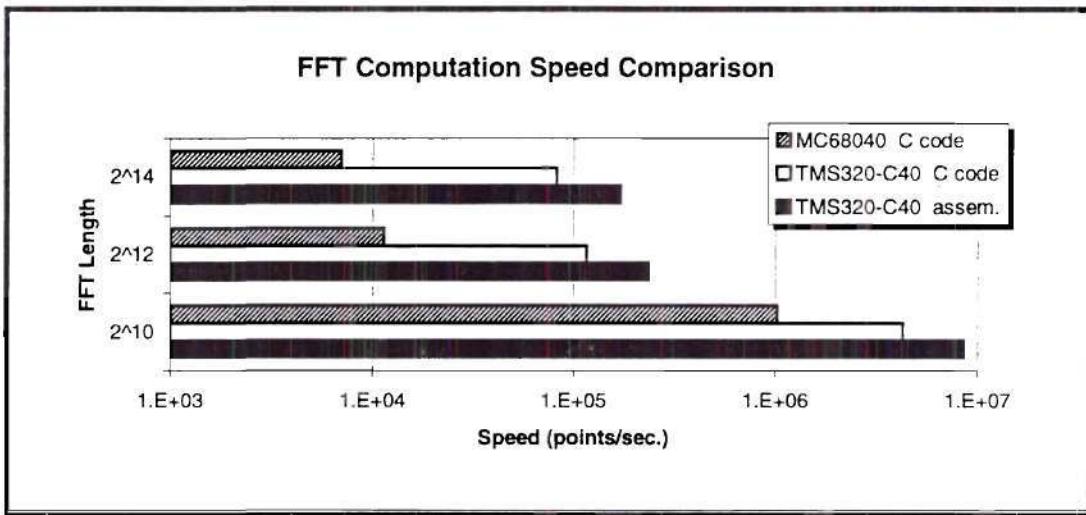


Figure 3.4 Performance Comparison of Hardware and Software

3.2.3 Handshaking and Buffering

In order to be implemented for on-line adaptive control, the frequency analysis software must contain sufficient buffering and reliable handshaking procedures to ensure that a continuous and appropriately segmented stream of data is available to the FFT code. In addition, flexibility was designed into the software to allow the greatest range of future applications.

Two examples of the flexibility of the handshaking system employed are listed here. First, the FFT width can be adjusted in real-time. This feature may become useful if it is necessary to expand and contract the FFT to accommodate peaks wandering to the edges of the Nyquist frequency range. However, because of the array storage techniques

that were employed in this work, the entire history buffer is zeroed each time the FFT width is changed, destroying any previously acquired data. A second feature of the software developed is that the number and size of the separate batches of time domain data being passed to the FFT can be adjusted in real-time, without disturbing the data buffered from previous batches. This capability can be used to maximize the computing power required of the combined CPU and DSP system.

Figure 3.5 shows a diagram of the handshaking scheme utilized for data exchange between the DSP board and the controlling CPU board. Blocks of memory on the DSP board were reserved using assembly language section commands. These memory block addresses were then saved in a header file to be linked to any C++ code on the CPU board desiring interaction with the DSP board. The most important of these memory blocks is where the new batches of time-domain data are written to by the CPU board, hereafter referred to as the batch memory block. After initialization of the DSP board, data integrity was ensured by requiring the DSP board to flag the batch memory block with a value outside of the dynamic range of the data being passed. The DSP code then periodically checked that memory location until it was overwritten by new data.

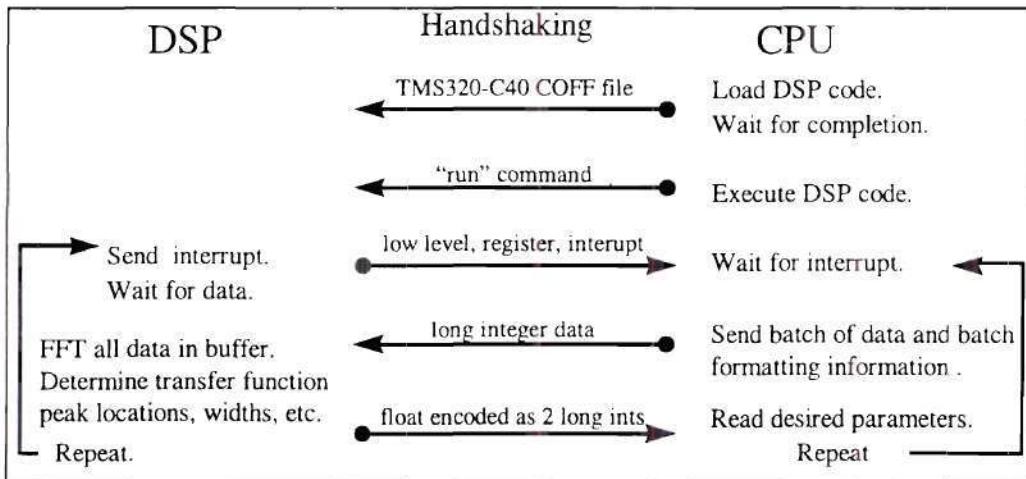


Figure 3.5 DSP-CPU Handshaking Schematic

Once new data was available for processing, it was transferred to a rolling buffer of data received previously. The assembly language FFT was then performed on the rolling memory block, which had been divided into appropriate sections for each input and output data record. This frequency domain information was then utilized to compute empirical transfer function estimates and determine the parameters of the poles of these transfer functions using the techniques discussed in Chapter II.

Once the pole parameters were computed and passed to a predetermined memory location, the batch memory block was flagged again as being empty and a low level interrupt was sent to the CPU board to immediately notify it that the DSP board was ready for more data. The typical computation time observed for straightforward implementation of the ETFE approach and simple global maximum search for a two-input, two output, four transfer function system was 100 ms. During this interim the CPU board was able to continue processing its feedback control loop as long as the batch size had been set large

enough to accommodate the computation time required on the DSP board. This is the main reason why iterative methods for parameter estimation on the DSP board were rejected. Their computation requirements are data dependent and could not be assured to be less than the time allowed for by the batch size. Once the CPU board had received the interrupt, and immediately before sending new data, it read the desired parameter estimates and utilized them to adjust the OATF filter coefficients, as required.

3.3 Identification and OATF Filtering during Feedback Control

A brief discussion of the particular implementation of the Optimal Arbitrary Time-Delay Filter for adaptive filtering control of RALF is discussed here.

3.3.1 Location of OATF Filter and Identifier in Feedback Control System

A diagram of the proposed feedback loop employed for identification, filtering, and control of RALF is provided here. Dotted lines indicate modules and connections that were removed or added during various experiments for comparison. Diagonal arrows through the OATF filtering blocks emphasizes their adaptive nature. The adaptive OATF filter was only employed in either the outer location or the inner location indicated, and was not employed in both locations simultaneously as was done for some experimental work by Magee [11].

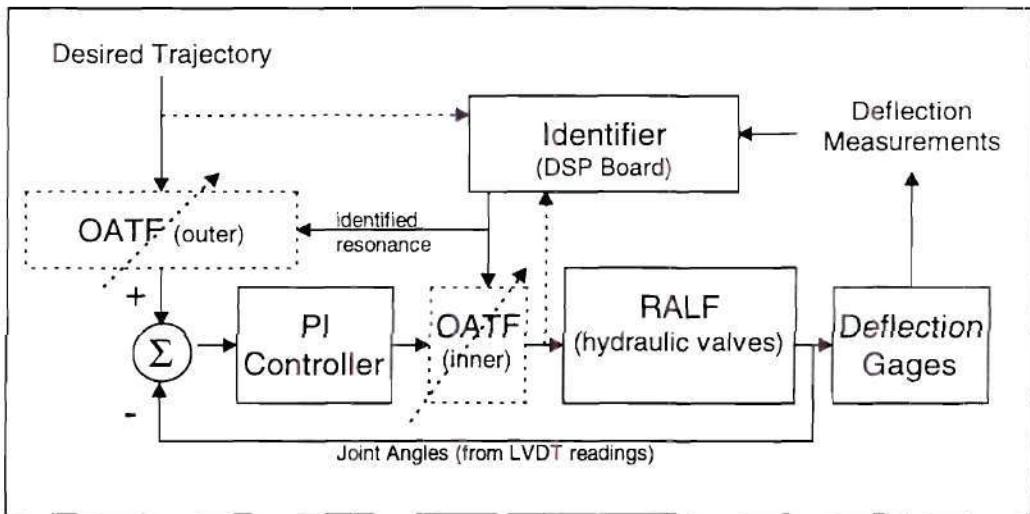


Figure 3.6 Possible OATF and Identifier Locations in Feedback Control of RALF

Comparing the feedforward configuration (the OATF in the outer location), with the feedback filtering configuration it can be seen that the feedforward configuration provides the advantage of not introducing time-delays within the feedback loop. This assures stability of the entire system provided the feedback loop is stable without the filter. In the feedback filtering configuration, the OATF filter may introduce instabilities not inherent to the unfiltered feedback loop. However, the feedback configuration has the advantage of filtering the signal fed to the actuators after all noise has been introduced into the signal. The feedforward configuration allows sensor noise, from the joint angle sensors, for example, to bypass the OATF and reach the feedback loop unfiltered. This noise problem proved to be a significant consideration in the simulation and experimental results of Chapter IV.

3.3.2 Discrete OATF Time-Delays

The details of the OATF filtering object described in Section 3.2.2 are discussed here because of the approximations required for discrete implementation of the continuous-time OATF equations. A single, OATF filter, normalized with for total gain of one, as developed by Magee and Book [11], is given by the impulse response

$$f_{t_{\text{opt}}}(t) = \frac{\delta(t) - 2 \cos(\omega_d T_1) e^{-\zeta \omega_n T_1} \delta(t - T_1) + e^{-2\zeta \omega_n T_1} \delta(t - 2T_1)}{1 - 2 \cos(\omega_d T_1) e^{-\zeta \omega_n T_1} + e^{-2\zeta \omega_n T_1}}, \quad (3-32)$$

where ω_d and ω_n are the damped natural frequency and the undamped natural frequency of the pole being filtered and ζ represents the damping ratio of that pole. The time delay, T_1 , is arbitrary and provides a means of trading increased response time for reduced residual vibration. The most robust value for vibration reduction and the value used by Singer and Seering is given by

$$T_1 = \frac{\pi}{\omega_d}. \quad (3-33)$$

When implementing the filter in a discrete-time system, the filter coefficients must be computed from a realizable discrete approximation of the time delay rather than the exact time delay given by Equation (3-33). The z-domain transfer function version of the filter is then

$$F_{i_{\text{opt}}}(z^{-1}) = \frac{1 - 2 \cos(\omega_0 n_i T_s) e^{-\zeta \omega_0 n_i T_s} z^{-1} + e^{-2\zeta \omega_0 n_i T_s} z^{-2}}{1 - 2 \cos(\omega_0 n_i T_s) e^{-\zeta \omega_0 n_i T_s} + e^{-2\zeta \omega_0 n_i T_s}} \quad (3-34)$$

where T_s is the sampling period, and n_i is an integer number of samples that approximates the delay time and is defined by

$$n_i = \text{int}\left(\frac{T_t}{T_s}\right). \quad (3-35)$$

where the `int()` function truncates a real number into an integer.

The importance of the proper discrete adaptation of the OATF filter given by Equation (3-34) can be seen in the following example. A three term filter for a system with a resonance at 51 Hz and a sampling rate of 500 Hz was designed improperly using Equations (3-32) and (3-33) directly. The resulting frequency response of the OATF filter is shown in Figure 3.7. Utilizing the correct integer multiple of samples for the time delay value results in the frequency response shown in Figure 3.8. The correct implementation always places at least one zero at the desired frequency, but higher frequency zeros will not necessarily be located at integer multiples of this frequency.

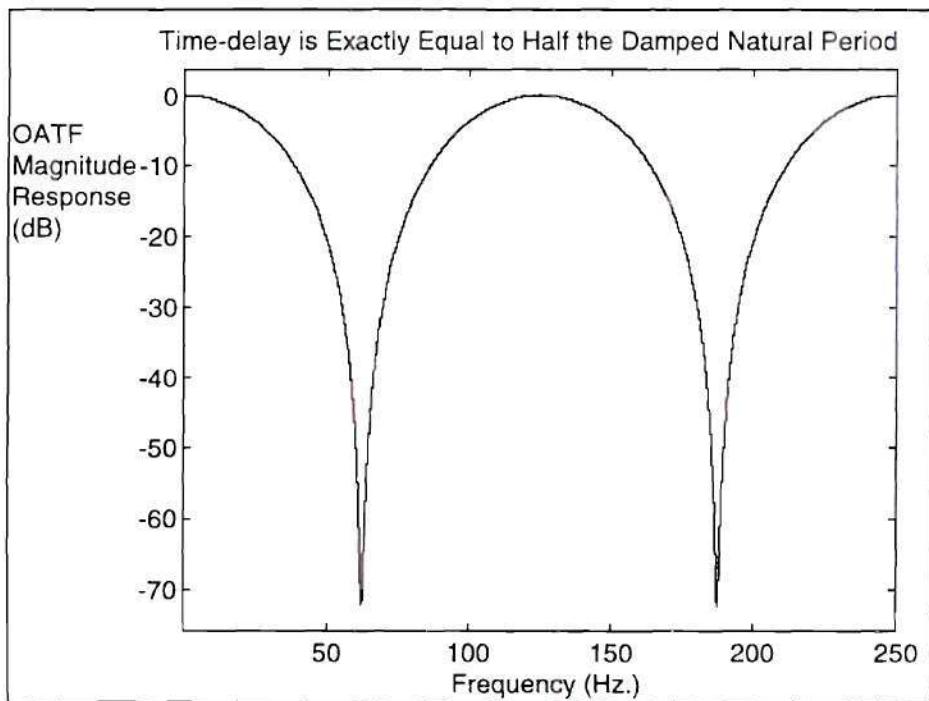


Figure 3.7 Improperly Designed OATF, Frequency Response

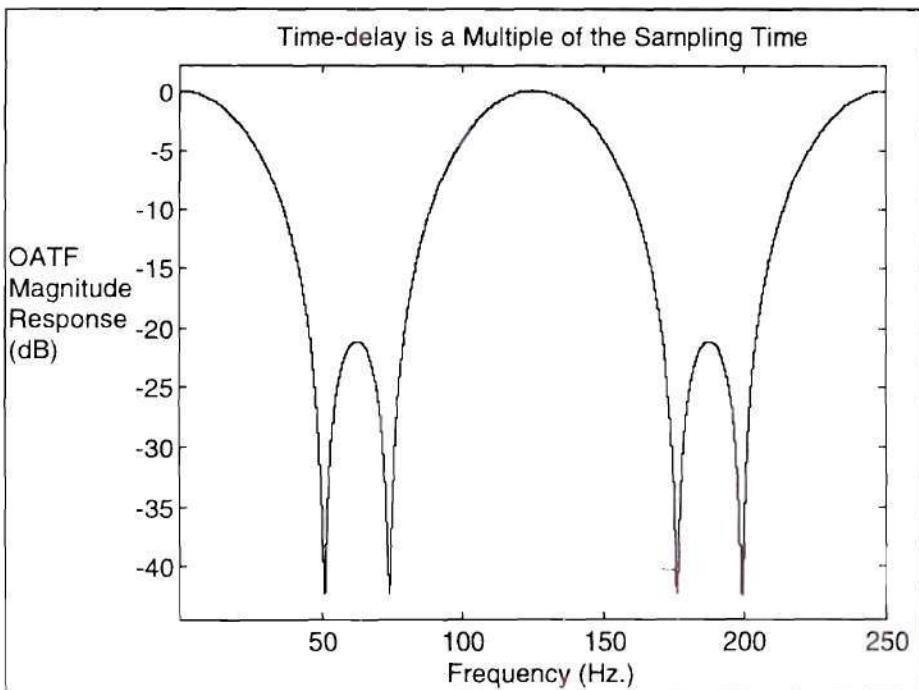


Figure 3.8 Correct Discrete Approximation OATF , Frequency Response

Direct computation of the filter coefficients from Equation (3-34) requires $O(23m_r + 2a_r + 6t_r)$ operations. However, taking advantage of several common terms and temporarily storing these values during evaluation, the number and complexity of the computations can be reduced by more than 60% to $O(10m_r + 2a_r + 2t_r)$. These operations are required each time a new modal frequency is identified and is utilized to modify the filter coefficients. This slight computational improvement, utilized for the C++ OATF object implementation, only became an issue because adaptation of the filter coefficients in real time was required.

Once the filter coefficients have been computed, actual implementation of the three-term, discrete OATF filter requires $O(N(3m_r + 3a_r))$ operations per sample, where N is the number of filtered modes or poles. This is the same as for any three-term, time-invariant, digital filter. The OATF filter class can filter any number of inputs to a device with any number of vibrational modes. The OATF object assumes that it is desirable to filter all of the resonant modes from all of the inputs to produce the filter outputs. This is the desired result for controlling any physical plant.

CHAPTER IV

RESULTS

The primary motivation for this research into modal parameter identification in real time was to implement an adaptive input filtering algorithm for RALF. This chapter presents the results of several experiments and simulations intended to reveal the capabilities and limitations of utilizing the proposed identification algorithm in conjunction with the OATF filtering method in an adaptive control loop.

4.1 DSP Implementation Verification

Before utilizing parameter estimation algorithms in an adaptive filtering control loop, it was necessary to first verify the transfer function estimation algorithm implemented on the DSP board. The results of two such tests are presented here. The first test validated the implementation of the identification algorithm, and the second verified the algorithm itself.

4.1.1 TFE Implementation Verification

First, verification of the C code on the DSP board was accomplished by duplicating FFT, PSD and TFE computation code with an off-line version in the form of a Matlab function. This was intended to verify all complex algebra performed on the DSP board but not to verify the algorithm's effectiveness. This test produced virtually identical transfer

function estimates from both the DSP board and the Matlab function except for rounding error differences in the low level background noise spectrum. Figure 4.1 plots these two spectra at an arbitrary time during the experiment described in Section 4.2.2.

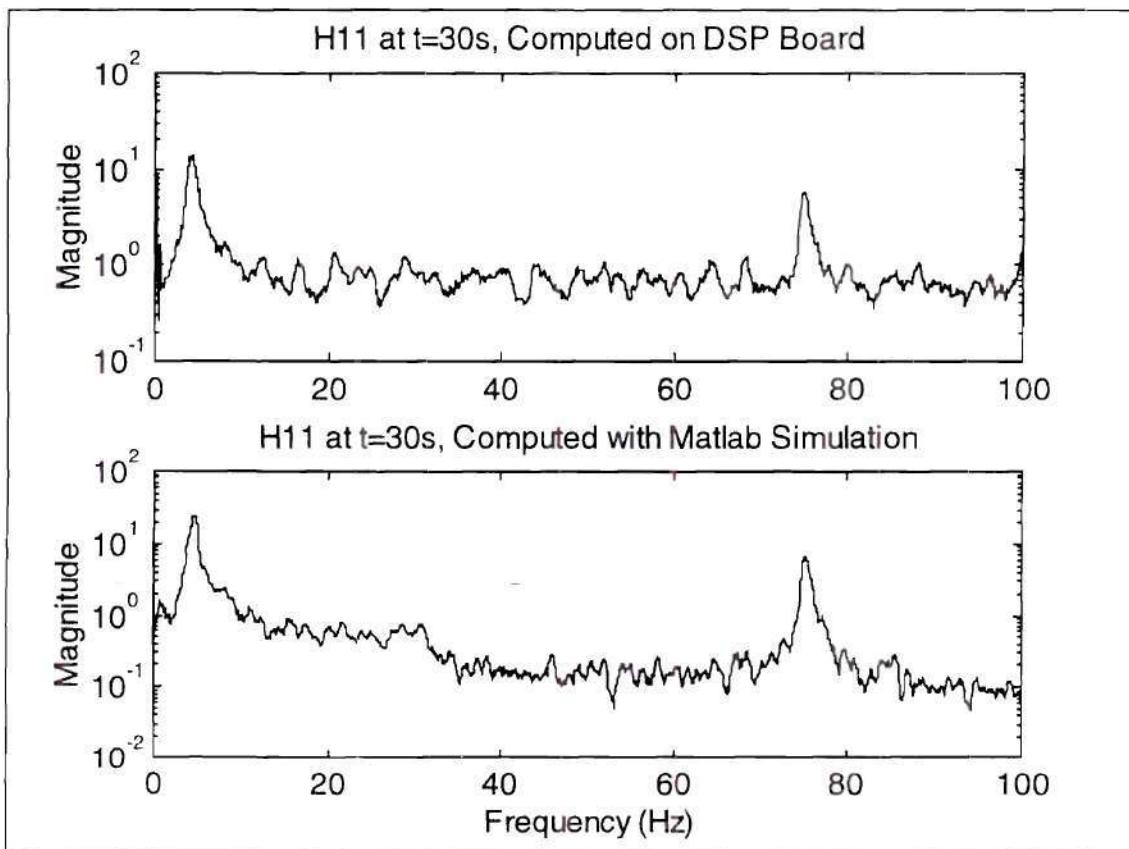


Figure 4.1 DSP Transfer Function Estimation Code Verification

4.1.2 Algorithm Verification

A second test was performed to verify the validity of the TTFE algorithm itself. Actual sensor and actuator data from the experimental trajectory were processed by both the DSP board and the commercially available Matlab spectral analysis function named

SPA(). SPA() utilizes the correlation matrix between inputs and outputs to determine transfer function estimates between those inputs and outputs. This method is a time-domain, least squares system identification technique and is considerably more computationally complex than the FFT-based system identification technique utilized for on-line identification in this research. In addition, SPA() subtracts out an estimated noise spectrum from its computed transfer functions. SPA() was intended for analysis of time-invariant systems and thus an approximation of the time-varying transfer function was accomplished by applying the SPA() function to limited ensembles of data evenly spaced in the time-domain record. These ensembles were aligned so that the transfer functions computed would correspond as closely as possible with those downloaded from the DSP board. Figure 4.2 compares the transfer functions estimated by the SPA() Matlab function and those computed on-line on the DSP board. As in Figure 4.1 a slight gain difference in the noise region is clearly visible, but the overall shape and location of the poles appears consistent. One noticeable difference is the greater degree of smoothing present in the DSP transfer function estimate. The transfer function estimation method used for this research required less than 150 ms to calculate on the DSP board while the Matlab SPA() function required 10 minutes on a 486-66 MHz PC.

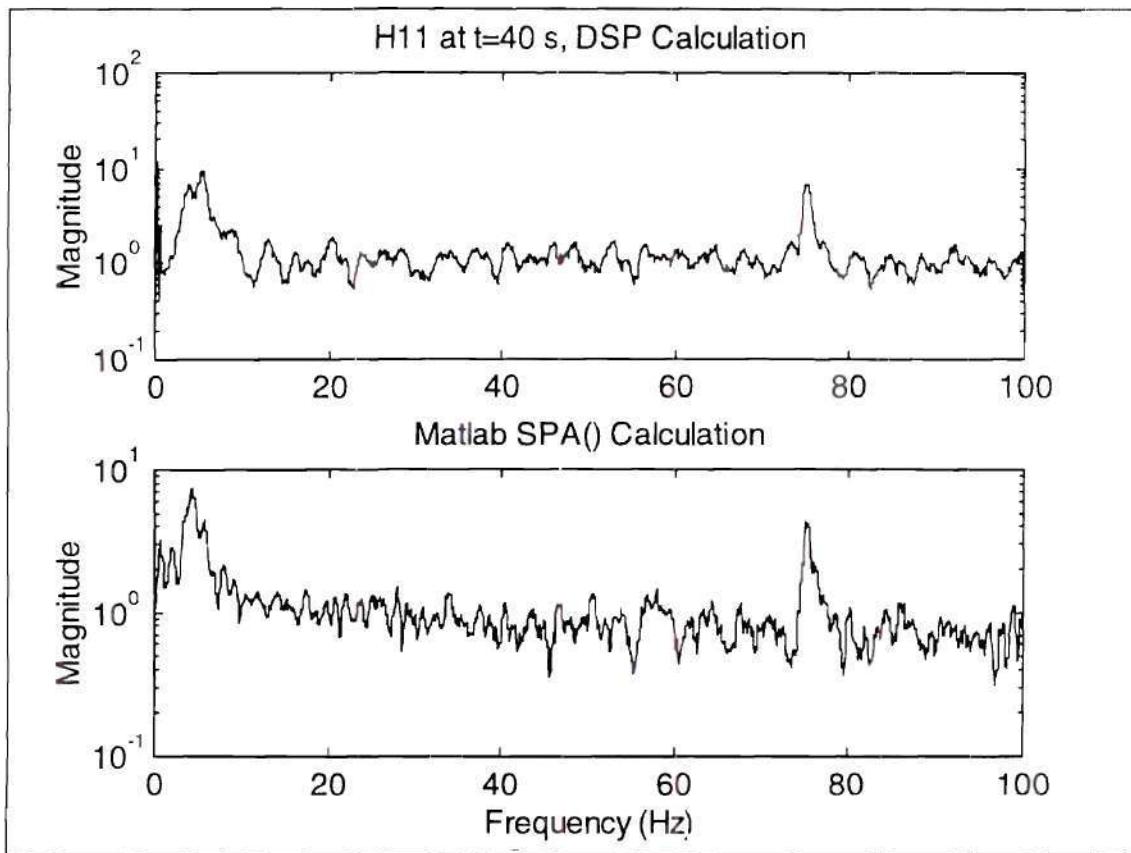


Figure 4.2 DSP Transfer Function Estimate Verification

4.2 Experimental Performance Comparison of Adaptive vs. Fixed OATF Filtering

Several experiments were performed to compare the performance of the fixed OATF filtering method with the adaptive OATF method developed for this research. Several features were common to all experiments and are discussed here. First, the desired tip trajectory is shown in Figure 4.3. The outer curved border indicates the workspace of RALF. The tip speed for all experiments was set to either 5 in/sec or 4 in/s as indicated, and the loop trajectory was performed in a counter clockwise direction. The trajectory was intended to simulated a weld or seam testing process. It was designed with right angles to

ensure vibration excitation. The squares in the trajectory each have a side of length 5 in. This results in an excitation with a period of 1 s for the 5 in/s tip speed used for most experiments. The pauses at either end of the first section of the trajectory were intended to allow vibration to settle before beginning each section.

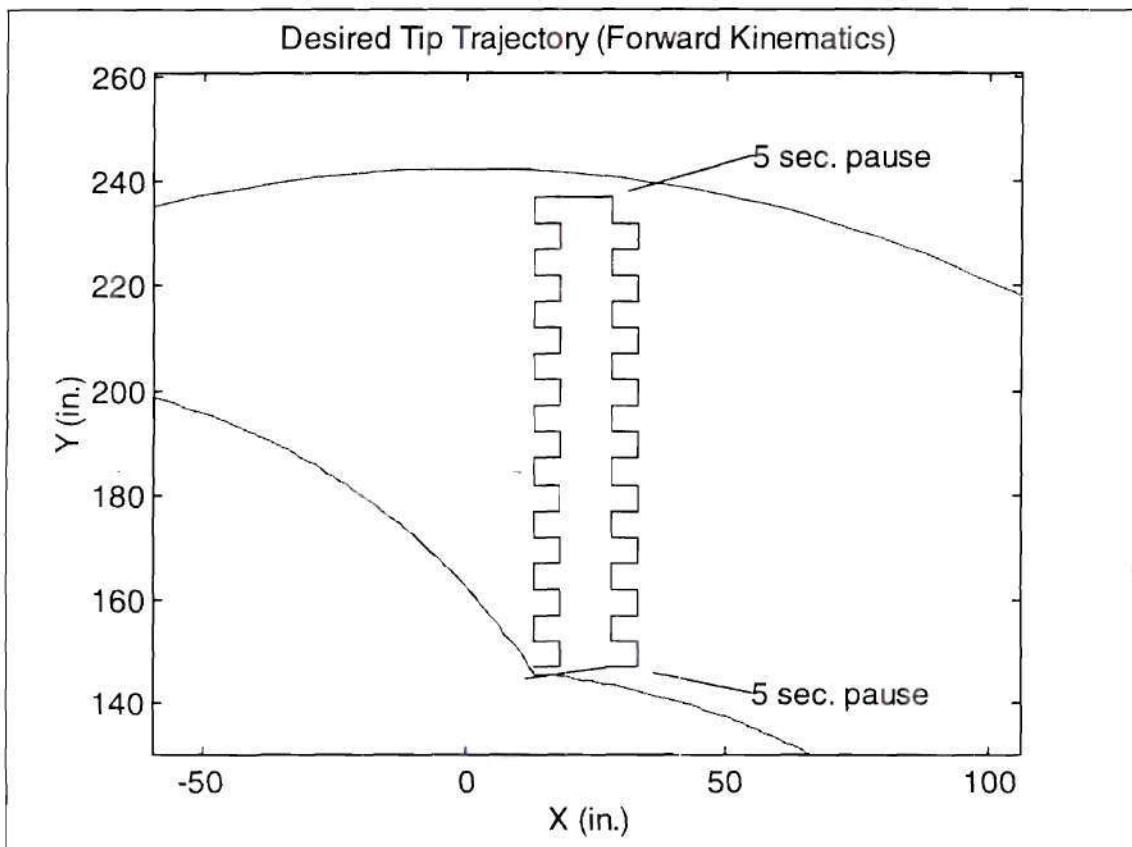


Figure 4.3 Desired Tip Trajectory

In all cases the sampling rate for both control and identification was 200 Hz. No down sampling was performed to allow control at a faster rate than identification. The FFT size chosen for all identification experiments was 2048. This sample rate and buffer length

provided a frequency resolution of .09766 Hz while not seriously hindering the identification of the time-varying parameters changing at a rate less than 5% a sec. The trade off between resolution and system tracking that FFT length selection provides is discussed in more detail by Oppenheim and Schafer [17] as well as Giordano and Hsu [6].

For the PI controller, proportional feedback gains from the angle error to the hydraulic actuator valve voltages of 7500 and 6250 mV/rad were selected for joint 1 and joint 2 respectively. Integral gains of 5000 and 4166 mV/rad·s were chosen. The integrator was implemented as a straight summer windowed exponentially with a forgetting factor of 0.0001, determined empirically to prevent the integral term from saturating the actuators. In addition, an integral threshold value of .04 radians was implemented to prevent integral summing during large error signals, though this threshold was never exceeded during the automated trajectory used for the results in this chapter. When transfer function information was downloaded from the DSP board, it was captured at regular intervals of 1000 samples (5 s) or 1200 samples (6 s) depending on the memory and computation requirements of the particular controller implementation.

4.2.1 Performance Evaluation Criteria

For comparison of the performance of the various filtering techniques two criteria were established. The first is the root mean square link tip velocity to be denoted RMS velocity. The second is the root mean square link tip acceleration and will be referred to as RMS acceleration. Both values were calculated by filtering the deflection measurements with customized linear differentiators. The velocity was calculated as the slope of a

piecewise linear regression with a window width of 10 samples, far less than the period of the expected oscillations. This smoothing differentiator provides greater than 20dB of attenuation for signal frequencies above 40 Hz when compared to a straight backward difference differentiator. The acceleration was computed as the second derivative of a 40 point piecewise quadratic regression. This provides more than 50 dB of attenuation above 20 Hz when compared to a backward difference approach. The root mean square, or standard deviation, of the velocity and acceleration signals was then computed as a cost function. Both the acceleration and the velocity signal exhibited the zero mean , zero drift characteristics desired.

Computation of an RMS link deflection value was problematic, requiring a piecewise detrending operation due to the considerable static deflection variation in both links. This detrending operation would have produced variable results dependent upon window alignment with static deflection profile features. As a result, it was not computed as a performance evaluator.

4.2.2 PI Controller

As a base-line for comparison of controller implementations the deflection measurements for a test run through the desired trajectory using a PI controller is provided in Figure 4.4.

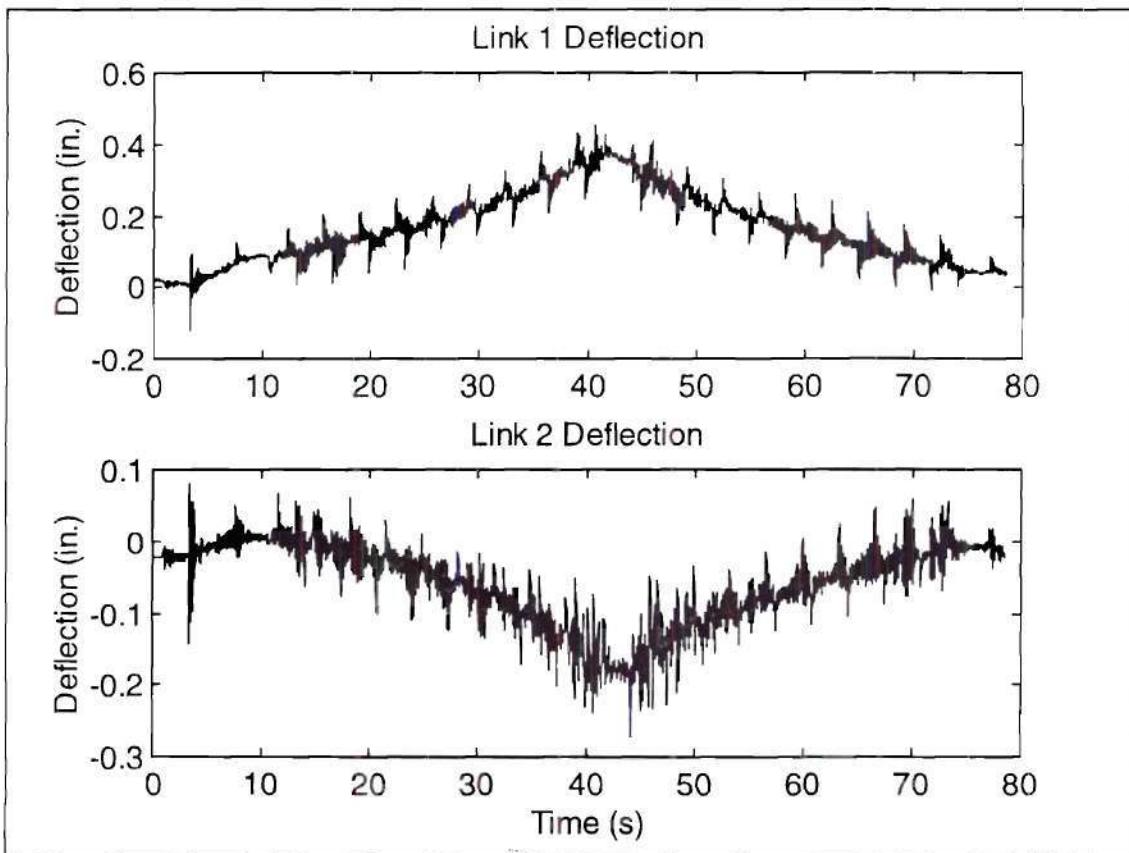


Figure 4.4 Link Deflection Under PI Control

Figure 4.5 shows a spectrogram of the PI control experiment from the transfer function estimates downloaded from the DSP board every 5 seconds. The transfer function labeling notation is the same as that used in Chapter 2, e.g. H_{12} denotes the transfer function from the first input to the second output. H_{tot} was constructed by combining all four transfer function estimates in a linear, unweighted average.

One interesting feature in Figure 4.5 is the split in the resonant peaks that occurs when RALF approaches a kinematic singularity and resembles the dynamic situation of an inverted double pendulum. This poses difficulty for any peak tracking algorithm. It can be

seen that the transfer function estimate heavily favors the higher frequency pole. To overcome this difficulty, in experimental work a logarithmic weighting function was utilized to favor lower frequencies during the global maximum search. This was deemed necessary because the low frequency pole is the one most likely to be excited by most practical trajectories including the ones considered here. Another interesting feature in Figure 4.5 is the delay between the system configuration change and corresponding changes in the estimated transfer function. Comparing Figure 4.4 and Figure 4.5 it can be seen that link 2 reaches its minimum deflection several seconds before the spectrogram shows the maximum separation in the identified poles.

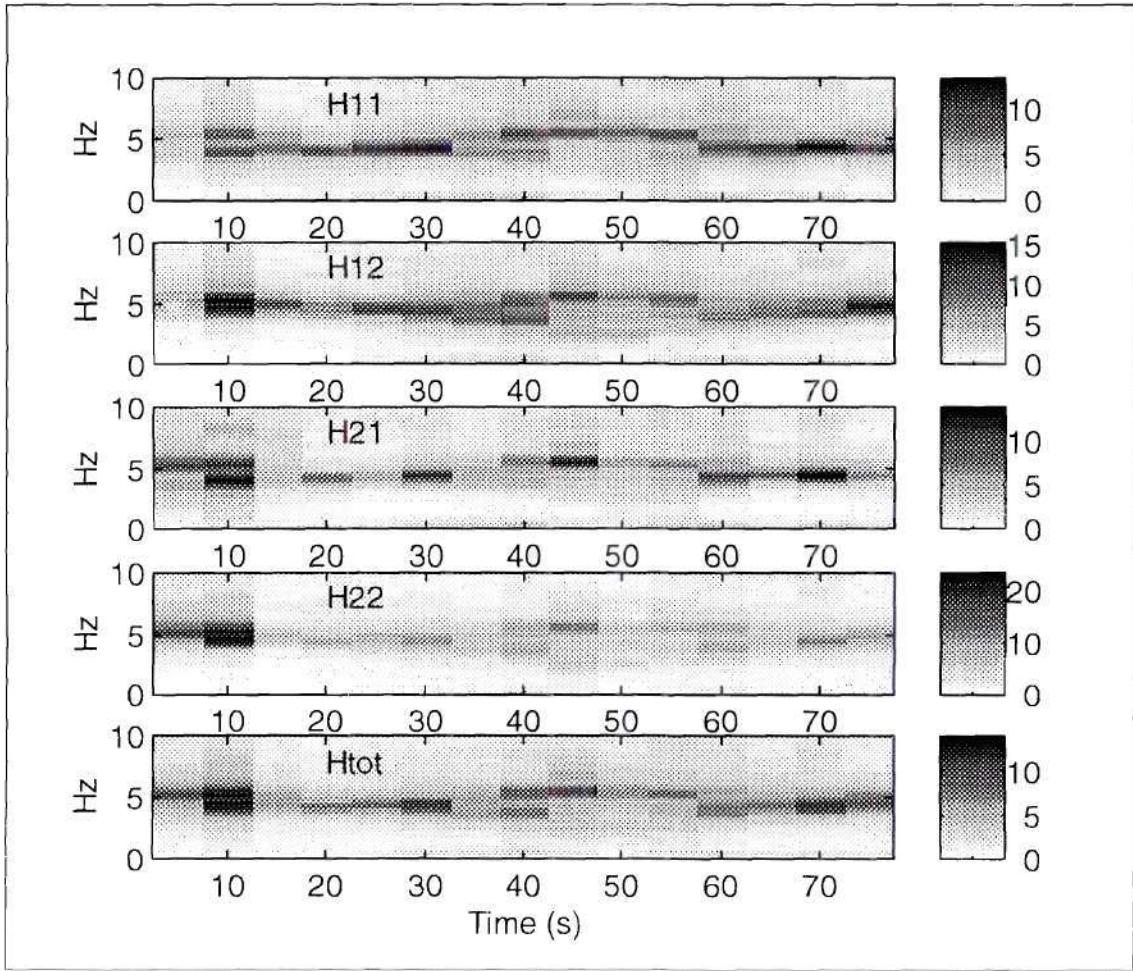


Figure 4.5 Spectrogram of PI Control Experiment

The Matlab function SPECGRAM() was utilized to produce Figure 4.6 and is provided for comparison with Figure 4.5. However, this Matlab function does not compensate for input power spectra as the ETFE and TTFE methods do. Thus the low frequency data is quite cluttered because of the large amount of low frequency excitation. Nonetheless, the splitting of the system poles at the apex of the trajectory is evident in this spectrogram, and it provides further confirmation of the delayed nature of the frequency information in Figure 4.5 when the two are compared.

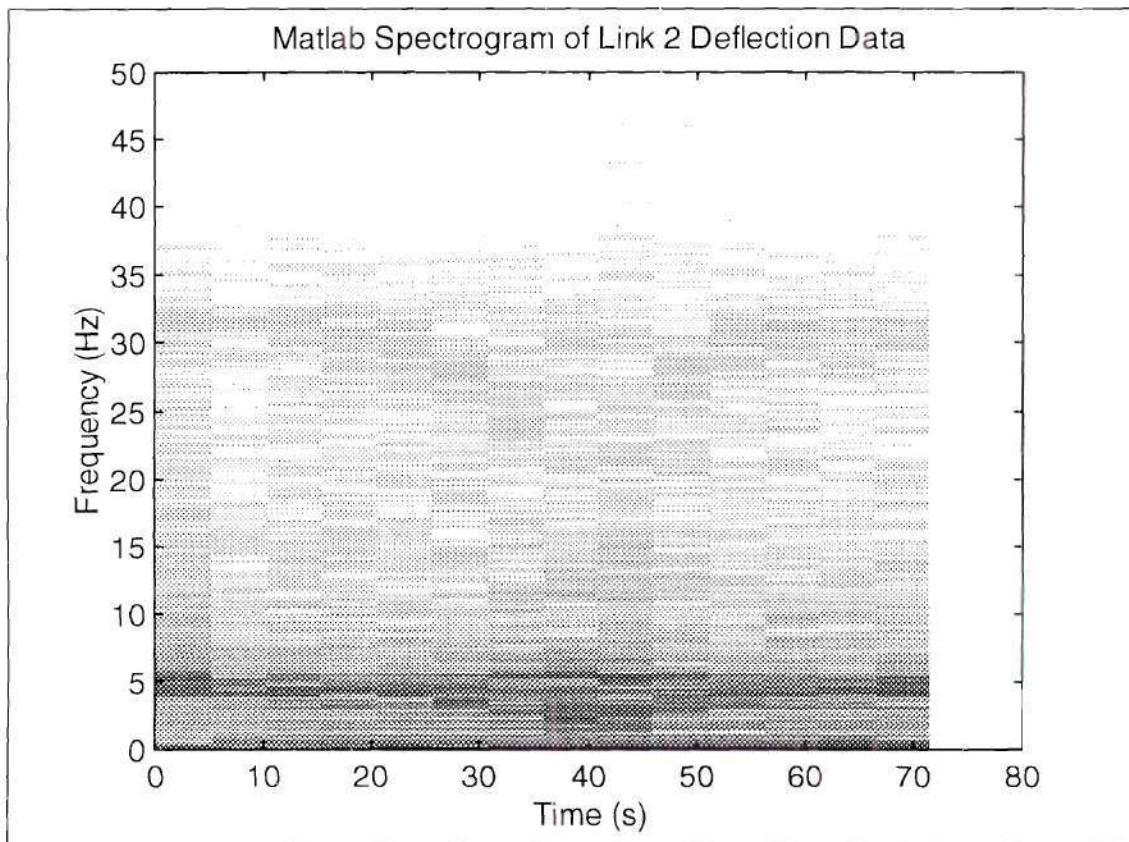


Figure 4.6 Matlab Spectrogram of Link 2 Deflection Data

4.2.3 Performance Comparison for Various Trajectories and Payloads

A battery of experiments was performed to determine whether the adaptive filtering technique could improve upon the performance of a fixed frequency OATF filtering technique. Five experiments were performed for two different trajectories and two payloads for a total of 20 test runs. The two trajectories were identical except of the commanded tip speed, either 4 in/s or 5 in/s. The payload that was removed and attached consisted of a steel plate weighing 15.65 lb. The five PI control experiments in each group

compared various filtering situations: a) unfiltered control; b) properly tuned fixed OATF filtering, c) properly tuned OATF filtering whose resonant frequency was allowed to adapt to changing system resonances; d) an improperly tuned fixed OATF filter; and e) an improperly tuned, fixed OATF filter that was allowed to adapt to the identified system resonance. The improperly tuned OATF was set to filter an arbitrarily chosen 10 Hz resonance. In all cases, the damping ratio of the OATF filter was fixed at .09, a value estimated by using the off-line FWHM approach on spectra acquired during an unfiltered trajectory run.

Initial results showed a significant reduction in RMS link velocity and acceleration when comparing a fixed, improperly tuned OATF filter to one allowed to adapt. However, as a weld began to separate in the base of the testbed, the observed improvement deteriorated. Only those results which were repeatable on the testbed as it now exists are presented here.

The values displayed in the left half of Table 4-1 are the RMS link deflection velocities and accelerations. Percentages are calculated based upon an arbitrarily chosen baseline experiment for easy comparison and are provided in the right half. From Table 4-1 it can be seen that an adaptive OATF filter provides no significant reduction in link deflection velocity and acceleration over a fixed OATF filter. Even when the OATF filter is intentionally tuned to an invalid damped natural frequency, on-line adaptation of the coefficients of the filter to more effectively filter the lower frequency resonance, does not improve performance. These were troubling results because it is known that modifying the OATF filter coefficients without modifying the time-delay values can indeed reduce

vibration when the initial OATF filter is chosen to filter an improper frequency.

Simulations follow which justify the poor performance of an adaptive OATF filter.

Table 4-1 RMS Velocity and Acceleration Over Entire Trajectory, Adaptive vs. Fixed OATF

Experiment 1: Payload Attached, Tip Speed = 5 in/s, Average $w_{n1}=3.6$ Hz.								
	Link 1 (in/s)	Link 2 (in/s)	Link 1 (in/s ²)	Link 2 (in/s ²)	Link 1 %	Link 2 %	Link 1 %	Link 2 %
a) Unfiltered	0.317	0.359	4.807	5.354	100.00	100.00	100.00	100.00
b) Fixed OATF	0.175	0.244	2.555	3.576	55.28	67.99	53.15	66.78
c) Adapt OATF	0.179	0.243	2.628	3.550	56.62	67.57	54.66	66.30
d) Poor OATF	0.228	0.288	3.427	4.287	72.03	80.34	71.28	80.06
e) Adapt Poor OATF	0.224	0.302	3.349	4.462	70.78	83.98	69.67	83.33
Experiment 2: Payload Removed, Tip Speed = 5 in/s, Average $w_{n1}=4.4$ Hz.								
	Link 1 (in/s)	Link 2 (in/s)	Link 1 (in/s ²)	Link 2 (in/s ²)	Link 1 %	Link 2 %	Link 1 %	Link 2 %
a) Unfiltered	0.262	0.273	3.833	3.573	82.67	76.15	79.73	66.73
b) Fixed OATF	0.165	0.190	2.374	2.386	52.20	52.90	49.38	44.57
c) Adapt OATF	0.169	0.197	2.407	2.468	53.28	54.81	50.08	46.08
d) Poor OATF	0.192	0.200	2.822	2.608	60.74	55.56	58.71	48.70
e) Adapt Poor OATF	0.210	0.216	3.109	2.854	66.44	60.13	64.68	53.30
Experiment 3: Payload Attached, Tip Speed = 4 in/s, Average $w_{n1}=3.9$ Hz.								
	Link 1 (in/s)	Link 2 (in/s)	Link 1 (in/s ²)	Link 2 (in/s ²)	Link 1 %	Link 2 %	Link 1 %	Link 2 %
a) Unfiltered	0.426	0.478	6.413	7.109	134.46	133.11	133.40	132.77
b) Fixed OATF	0.184	0.254	2.552	3.580	57.96	70.69	53.08	66.86
c) Adapt OATF	0.188	0.273	2.634	3.897	59.23	76.02	54.80	72.78
d) Poor OATF	0.236	0.317	3.424	4.569	74.36	88.23	71.21	85.33
e) Adapt Poor OATF	0.233	0.318	3.384	4.575	73.43	88.64	70.39	85.43
Experiment 4: Payload Removed, Tip Speed = 4 in/s, Average $w_{n1}=4.1$ Hz.								
	Link 1 (in/s)	Link 2 (in/s)	Link 1 (in/s ²)	Link 2 (in/s ²)	Link 1 %	Link 2 %	Link 1 %	Link 2 %
a) Unfiltered	0.358	0.353	5.177	4.613	113.06	98.21	107.69	86.16
b) Fixed OATF	0.214	0.236	2.965	2.918	67.46	65.73	61.67	54.50
c) Adapt OATF	0.208	0.239	2.881	3.016	65.59	66.59	59.92	56.33
d) Poor OATF	0.226	0.220	3.284	2.879	71.31	61.25	68.30	53.77
e) Adapt Poor OATF	0.221	0.228	3.178	2.962	69.75	63.54	66.10	55.32

4.3 Simulation

One possible reason for the lack of improvement between adaptive and fixed OATF filtering observed in the experimental work of the previous section was determined to be the low resolution of the Temposonics transducers used for feedback of the joint angles in the PI control loop. To evaluate this possibility Matlab Simulink simulations were performed.

4.3.1 Model

A simplified velocity feedback control model of RALF was assumed. This simplification is possible due to the fact that the lightweight structure of RALF is dramatically overpowered by the 1000 psi hydraulic cylinders. The actuators reach a steady state continuous velocity in less than 50-milliseconds after an actuation signal is applied. This is on the order of the sampling rate for the experiments performed. The sampling rate for the controller was identical to experimental work and an appropriate integration sample time was chosen for the simulation.

The dynamics of the link elasticity were modeled as a single degree of freedom, second order, mass-spring-damper system. Approximations of the damping ratio and natural frequency were taken from experimental observations of RALF. For the simulations the link deflection transfer function system was given poles to correspond with a 5 Hz resonance and a damping ratio of 0.09. The same angle trajectories used for automated control of RALF were input into the simulated controller. The first simulation group was run without any noise added to any of the signals. In the second simulation

group, a band-limited white noise signal with a power of 0.2% of the trajectory signal was added to the filtered trajectory. The standard deviation of the noise signal was 1.41×10^{-4} rad, less than half the average resolution of 3.5×10^{-4} rad. and standard deviation of 3.9×10^{-4} rad. of provided by the Temposonic transducers, after sampling with 12 bit A/D board, used to measure RALF's joint angles.

4.3.2 Results

Table 4-2 provides simulation results in a similar format to the experimental results of Section 4.2.3. A bar chart of the deflection velocity values is provided in Figure 4.7. The noise free simulation shows that tuning an OATF filter can indeed have a significant effect on system performance. However, the second simulation results show how a small noise signal added to the filtered controller input drastically reduces not only the absolute vibration reduction of the filter, but also the relative differential between a well tuned OATF and a poorly tuned one. In addition, in the simulation the time delay values of the OATF filter were chosen as the most robust for the frequency being filtered. Such a luxury is not available to real-time adaptation. As a result, the third simulation employed a fixed delay time similar to the one used in experimental work, .05 s. Finally, a fourth simulation attempted to determine if reducing the damping ratio would increase the differential between the results for an OATF filter tuned to different frequencies. However, it was not expected to improve the performance over an OATF filter with a damping ratio matching the system.

Table 4-2 Simulation: Performance Differential Gained by Tuning OATF

Simulation 1: Ideal Case, Noise Free Input				
	RMS Deflection Velocity (in/s)	RMS Deflection Acceleration (in/s ²)	RMS Deflection Velocity %	RMS Deflection Acceleration %
Unfiltered:	0.890	23.594	100.00	100.00
Correct 5 Hz OATF	0.260	6.554	29.17	27.78
10 Hz OATF	0.451	10.838	50.65	45.93
50 Hz OATF	0.855	22.821	96.13	96.72
Simulation 2: Noise Added to Filtered Controller Input				
	RMS Deflection Velocity (in/s)	RMS Deflection Acceleration (in/s ²)	RMS Deflection Velocity %	RMS Deflection Acceleration %
Unfiltered:	1.252	26.552	140.66	112.53
Correct 5 Hz OATF:	0.923	15.640	107.00	66.29
10 Hz OATF:	0.994	16.724	111.74	70.88
50 Hz OATF:	1.067	19.620	119.87	83.16
Simulation 3: Noisy Input, Time Delay Value Fixed at .05 s				
	RMS Deflection Velocity (in/s)	RMS Deflection Acceleration (in/s ²)	RMS Deflection Velocity %	RMS Deflection Acceleration %
Correct 5 Hz OATF:	0.952	14.450	103.75	61.24
10 Hz OATF:	0.993	16.718	111.66	70.86
50 Hz OATF:	1.218	25.776	136.91	109.25
Simulation 4: Noisy Input, Fixed Time-Delay, Damping Ratio = 0.01				
	RMS Deflection Velocity (in/s)	RMS Deflection Acceleration (in/s ²)	RMS Deflection Velocity %	RMS Deflection Acceleration %
Correct 5 Hz OATF:	0.923	14.464	103.72	61.30
10 Hz OATF:	0.992	16.654	111.53	70.58
50 Hz OATF:	0.992	16.642	111.48	70.53

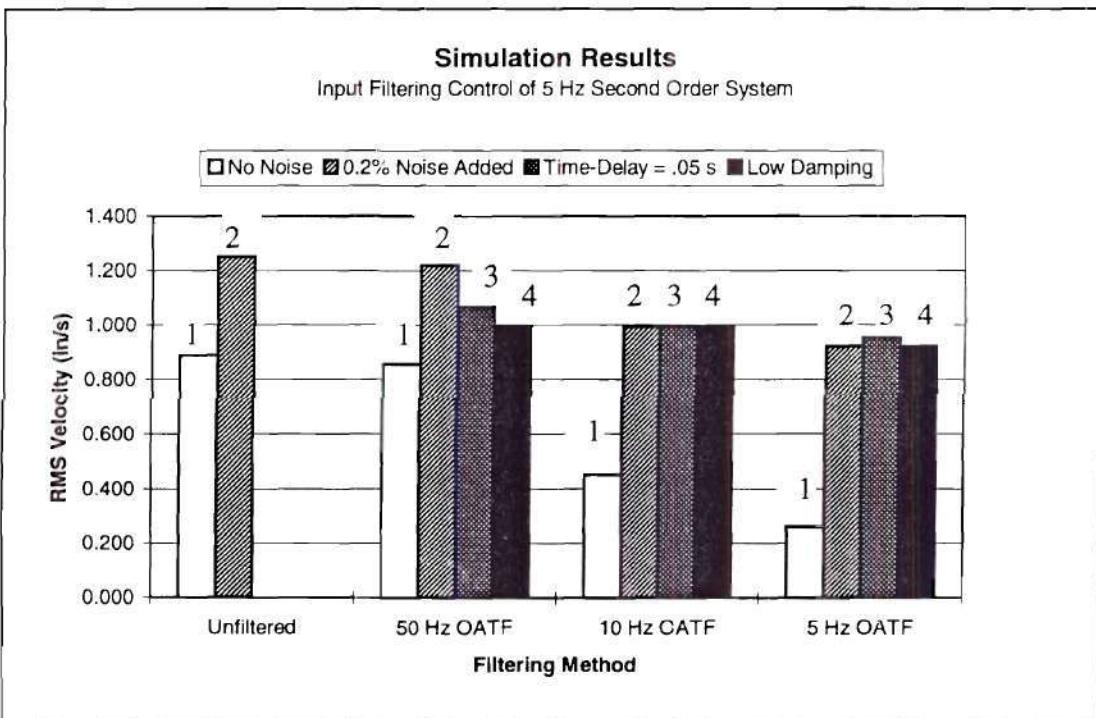


Figure 4.7 Simulation: Performance Differential Obtained by Tuning OATF

The improved OATF performance for a smaller damping ratio when utilizing a time-delay not equal to the Singer and Seering (most robust) value, can be explained by looking at the frequency response plot of the OATF filter for the filters implemented in Simulations 2, 3 and 4. Figure 4.8 shows that an OATF filter tuned to filter a lower damping ratio pole actually attenuates a sinusoid more at the resonant frequency than an OATF filter with a damping ratio that matches the pole.

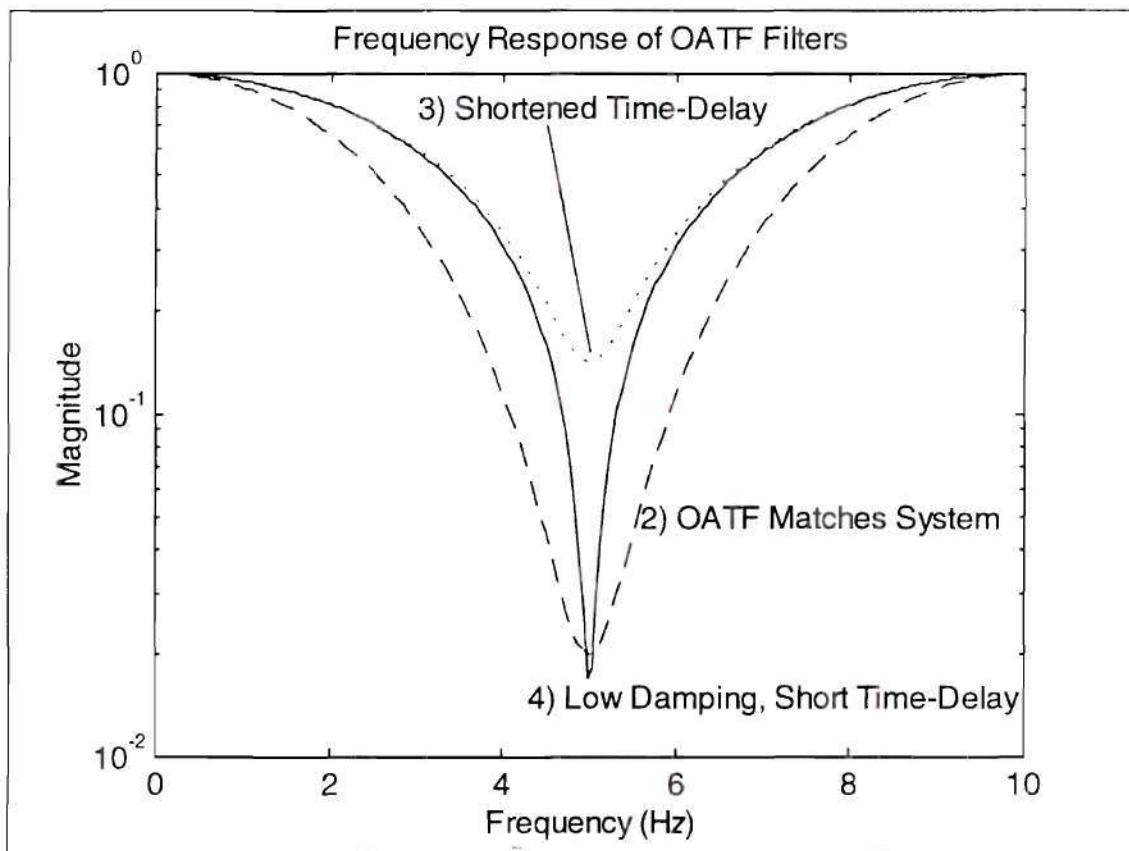


Figure 4.8 OATF Frequency Response for Reduced Damping Ratio

To reaffirm that the noise to signal ratio was not excessive, a plot of the desired trajectory along with the filtered trajectory plus the added noise is provided in Figure 4.9.

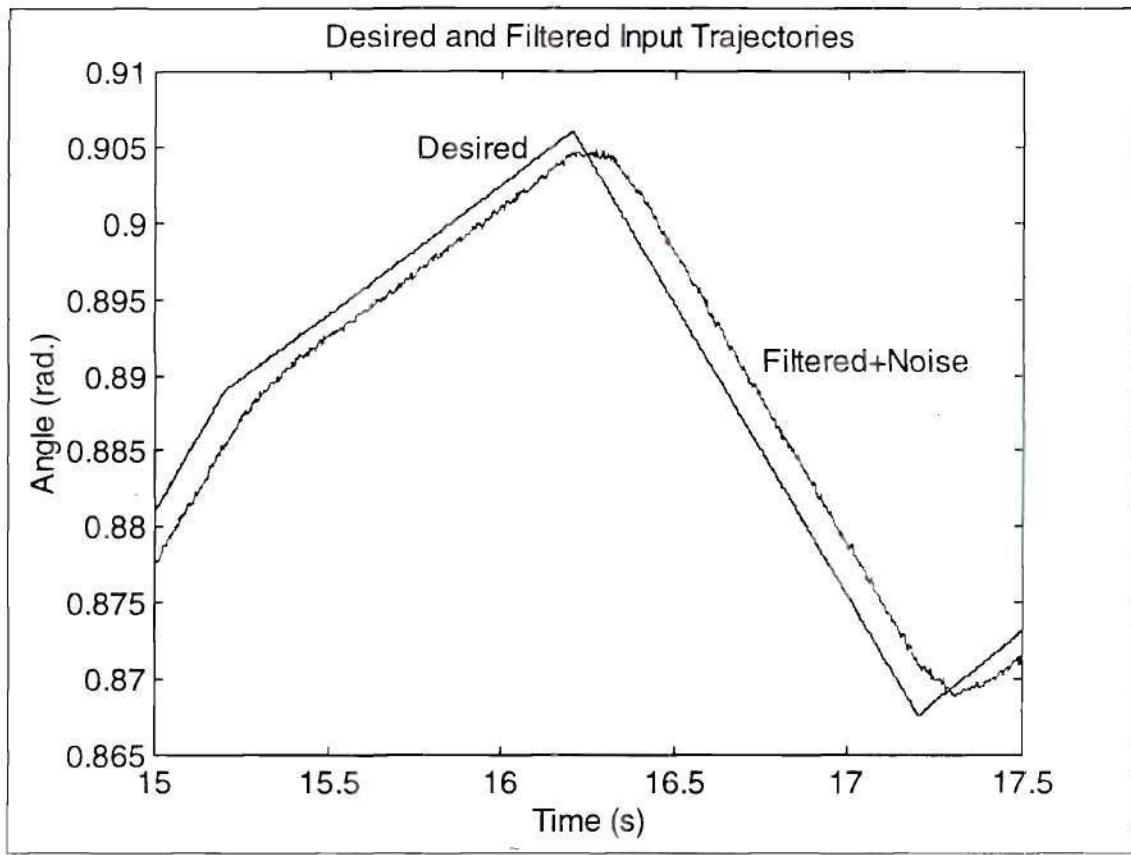


Figure 4.9 Desired Trajectory and Filtered Trajectory Plus Noise

4.4 Reduced Damping OATF Filtering Experiment

In order to increase the differential between the performance of a well-tuned OATF and one filtering a frequency different from the system resonance, it is advantageous to decrease the damping ratios of both filters. This effectively narrows the zeros applied by the filters and makes filter performance more sensitive to the design frequency. This of course reduces the robustness of the filters to error in resonant frequency. Experiments similar to those presented in Section 4.2.3 were performed with a smaller damping ratio of 0.01 used for the OATF filter. Figure 4.10 compares the deflection measurements of an

adaptive and nonadaptive OATF in the region near the singularity at the top of RALF's workspace. The unconnected points in the figure represent link deflections when the OATF filter was fixed at 10 Hz. The solid line shows deflection measurements during the adaptive filtering run. The results were encouraging, showing a greater than 30% reduction in deflection velocity of both links. This was consistent with simulations for low damping ratios. This confirms that the algorithm is indeed tracking and filtering the first fundamental resonance of the system. Nonetheless, when averaged over the entire trajectory, the RMS deflection velocity improvement reduces to 2%. This can be seen in Figure 4.11. In addition, a fixed OATF filter with an average value for the damping ratio and natural frequency still performs much better than an adaptive version with a lighter damping ratio.

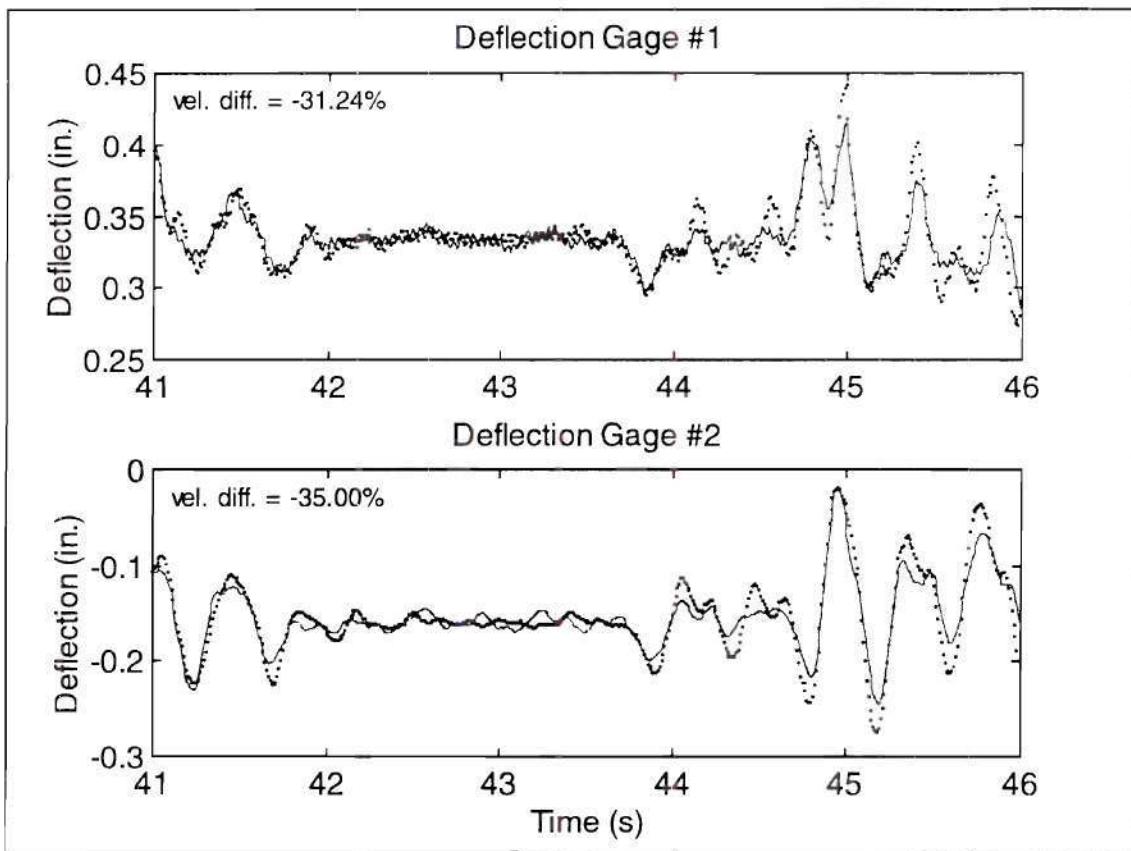


Figure 4.10 Deflection Measurements During Adaptive Filtering Near Singularity

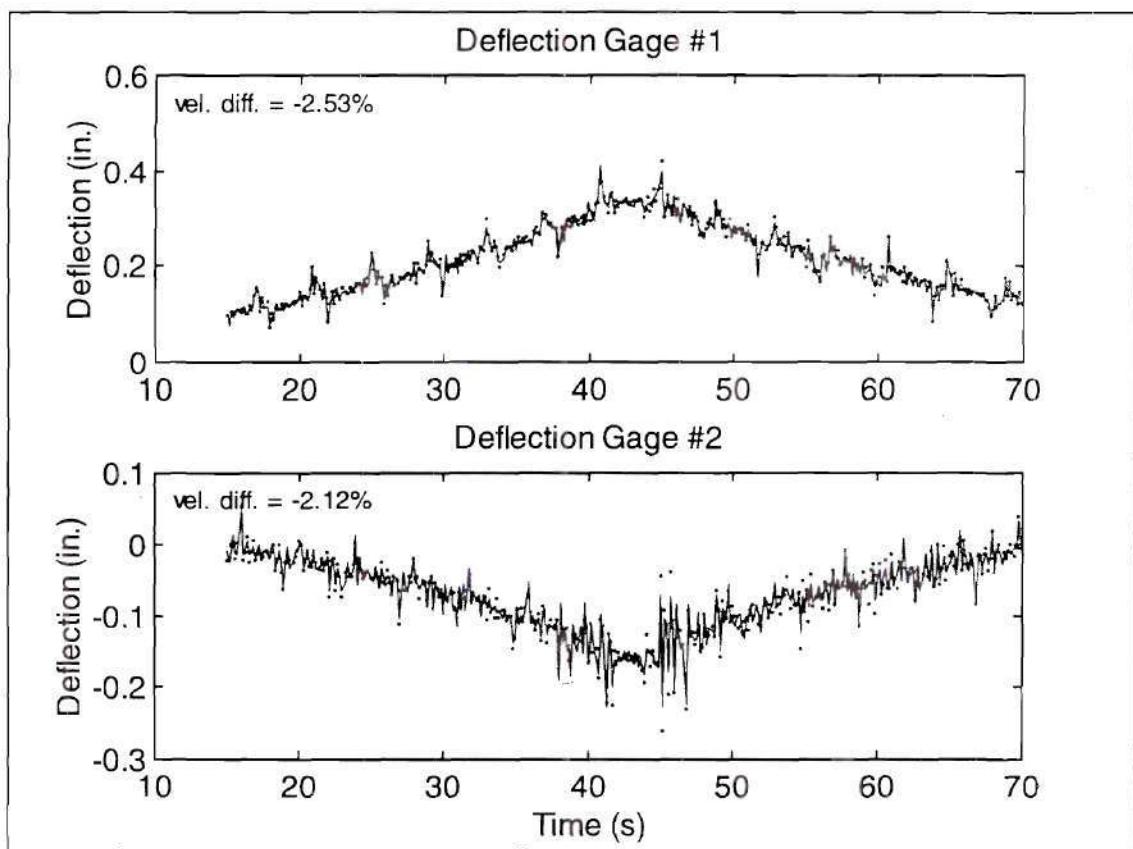


Figure 4.11 Deflection Measurements Over Entire Trajectory While Adaptive Filtering

CHAPTER V

CONCLUSION

This chapter will evaluate the results presented in the previous chapter. In addition the contributions of this research will be listed, and suggestions for future work are provided.

5.1 Evaluation of Results

5.1.1 Poor Adaptive OATF Performance

The small differential between adaptive and fixed OATF filtering for many of the experiments and trajectories discussed in Chapter IV was traced to several causes. First the identifying algorithm employed produces lag times of between one and four seconds between system parameter changes and changes in the identified modes. Higher frequency transfer function features exhibited a shorter lag time than those at the lower end of the spectrum, while the lower frequency modes were the most critical to filter performance.

Secondly, the presence of unfiltered modes necessarily reduces the effectiveness of any filter. In addition it reduces the differential between filters tuned to different

frequencies. Not only was the pole split ignored during the filtering of RALF's inputs, but the higher frequency poles that could not be identified at all were also ignored.

In addition, simulation results pointed out the importance of noise when comparing the performance of various tunings of an OATF filter. Binary noise, such as that resulting from the low resolution of the VDTs utilized to calculate RALF's joint angles, imparts the maximum possible energy to a system for a given amplitude [30]. It was shown that allowing any significant noise to bypass the filter and influence the actuation signal reduced the performance of the OATF filter and drastically reduced the differential between an improperly tuned OATF and one designed precisely to match the resonance of the controlled system. This provided a limiting case for the performance expected in experimental results for implementation of the adaptive OATF filter. The ideal performance of an adaptive OATF filter would result from a filter that exactly tracked the changing system parameters without any delay. Such a filter cannot perform any better than a fixed filter properly tuned to the resonance of a time-invariant system.

5.1.2 Possible Advantages of Adaptive OATF Filtering

Nonetheless, for OATF filters designed for lower damping ratios it was found that some improvement in system performance over a fixed OATF filter was possible. This was only possible for an OATF filter mistuned by 100% or more. However, smaller damping ratios are expected to have the added benefit of better joint trajectory tracking, though no such comparison was attempted in this research. Thus, if trajectory tracking is of critical importance, and system parameter changes are large, further investigation into

adaptive OATF filtering with lowered damping ratios would be reasonable. If, however, reduced link vibration is the goal, then a fixed OATF filter can achieve greater performance for parameter changes less than 100%.

In addition, if a robotic device must operate without human intervention for extended periods of time while its system parameters are expected to change drastically, it would be advisable to implement some sort of adaptation of the filter coefficients, and possibly even the time delays. However, to adapt the time delays during continuous operation would require investigation into the stability considerations that would result.

5.2 Contributions of This Research

A significant contribution of this research is the algorithm developed for online system identification. The algorithm and the results obtained from it and presented in this thesis may aid future researchers interested in similar problems. This thesis has revealed the limitations and capabilities of the various parameterization and transfer function estimation techniques investigated in Chapter II. This thesis also corrected typographical errors in published formulae used for system identification and clarified the origins of the techniques employed in TTIE. This body of knowledge may prove invaluable to future applications of online identification.

In addition, the increased understanding of the limitations of adaptive OATF filtering techniques gained by implementing it in a situation where noise considerations were paramount should not be underestimated. By showing in both experimental and simulation results what an adaptive OATF filter is capable of, this thesis may prevent

future researchers from attempting similar implementations before fully investigating the theoretical limitations of such an idea. Inversely related to the decreased level of expectations for adaptive OATF filtering is the increased appreciation for the robustness inherent in fixed frequency OATF filtering. The dramatic reduction in link vibration achievable by implementation of an OATF filter with fixed coefficients and time-delays is not easily surpassed. Perhaps future researchers will now be more willing to apply OATF filtering to applications where precise, time-invariant linear system models are not available.

Lastly, this research resulted in development of a vast library of C++ software developed for control, system identification, adaptive filtering, and real-time massive data buffering and storage. Because these modules were encapsulated in C++ objects, they will be more easily reused in future research projects. The extensive investment in additional software development time required to write and debug C++ code will pay off handsomely if the algorithms prove useful for other applications and researchers.

5.3 Future Work

The degree to which adaptive input filter performance depends upon noise free sensor feedback and how much it depends upon the delay between parametric changes and identification of those parameters should be investigated. One possibility would be to store the parametric information acquired during an initial pass through the workspace and then utilize that information at a later time to anticipate parameter changes with he filtering

algorithm. In addition, by improving sensor resolution significantly, this source of error could be reduced. In the case of RALF this would require upgrading an A/D board with 16 or 20 bit resolution.

Discontinuous parameter changes should also be implemented. This would allow more precise quantification of the delay time associated with any identification algorithm. It would also improve the system's applicability to the practical task of retrieving and releasing a payload.

Furthermore, more effective and reliable techniques for tracking multiple poles should be implemented. This would require increasing the resolution of the transfer function identifier. In addition it would be necessary to implement a sophisticated algorithm to determine when an additional filter zero should be created or an existing one eliminated. Increased transfer function resolution might also permit the identification of a damping ratio and adjusting the damping ratios of the filters to more closely match the poles.

5.4 Conclusion

This thesis failed in its goal of producing a widely applicable adaptive filtering technique to improve upon existing techniques for multi-degree of freedom devices. However, the techniques illustrated may indeed prove useful in a variety of situations such as those described in Section 5.1.2. In addition, this research provides an invaluable first step along the branch of control research that attempts to overcome nonstationarity with adaptive technologies.

APPENDIX A

KINEMATICS OF RALF

A.1.1. Forward Kinematics

The forward kinematics transformation for RALF was constructed by adhering to the Denavit-Hartenberg convention for robotic coordinates. Figure A.1 defines the joint angles and link dimensions. Additional angles and dimensions are shown for utilization in other references to the same device. Table A-1 lists the Denavit-Hartenberg parameters used to produce the transformation matrices for each link that were multiplied to produce the following total coordinate transformation from base to tip,

$$T_0^2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{A.1})$$

Table A-1 Denavit-Hartenberg Parameters for RALF

Link	a_i	α_i	d_i	θ_i
1	$a_1 = \sqrt{b_1^2 + b_2^2} = \sqrt{120^2 + 5.75^2}$ in.	0	0	θ_1
2	$a_2 = 120$ in.	0	0	θ_2

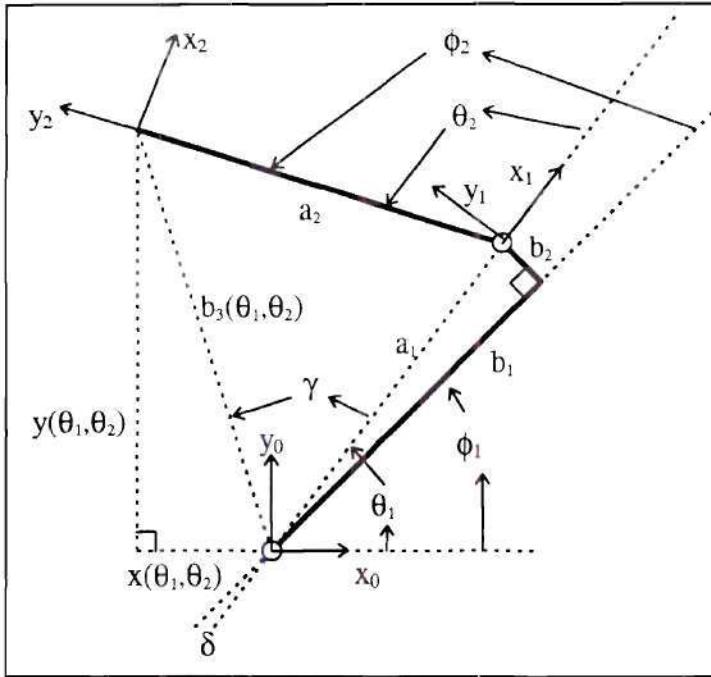


Figure A.1 Kinematics Diagram of RALF

A.1.2 Inverse Kinematics

Inverse kinematics is the problem of determining the joint angles that will produce the desired tip position for a rigid mechanical device. Because of the simple, planar kinematics of RALF, a geometric analytical solution to the inverse kinematics problem was possible. This method, of course does not take into account link deflection and serves only as a first approximation. Equation (A.2) is derived directly from the diagram. Equations (A.3) and (A.4) result from the law of cosines applied to the interior triangle formed by a_1 , a_2 and b_3 .

$$\theta_1 = \pi - \text{atan} 2(y_{tip}, x_{tip}) - \gamma \quad (\text{A.2})$$

$$\cos(\pi - \theta_2) = \frac{a_1^2 + a_2^2 - b_3^2}{2a_1a_2} \quad (\text{A.3})$$

$$\gamma = \cos^{-1} \left(\frac{a_1^2 + (x_{tip}^2 + y_{tip}^2) - a_2^2}{2a_1\sqrt{(x_{tip}^2 + y_{tip}^2)}} \right) \quad (\text{A.4})$$

These equations are then solved for θ_1 and θ_2 in terms of x_{tip} , y_{tip} and known dimensions while only considering the elbow-up posture. This is the only pair of solutions available within the joint limits of RALF. The first joint is capable of angles between 60° and 100° due to the stroke limit of the first hydraulic actuator. Similarly, solutions for the second joint are limited to angles between 10° and 120° .

$$\theta_1 = \pi - \text{atan}2(y_{tip}, x_{tip}) - \cos^{-1} \left(\frac{a_1^2 + (x_{tip}^2 + y_{tip}^2) - a_2^2}{2a_1\sqrt{(x_{tip}^2 + y_{tip}^2)}} \right) \quad (\text{A.5})$$

$$\theta_2 = \pi - \cos^{-1} \left(\frac{a_1^2 + a_2^2 - x_{tip}^2 - y_{tip}^2}{2a_1a_2} \right) \quad (\text{A.6})$$

However, if the ϕ 's are the joint variables of interest, the variables being controlled, they can be easily calculated as

$$\phi_1 = \theta_1 - \delta \quad (\text{A.7})$$

and

$$\phi_2 = \theta_2 + \delta \quad (\text{A.8})$$

where

$$\delta = \tan^{-1} \left(\frac{b_2}{b_1} \right). \quad (\text{A.9})$$

APPENDIX B

FULL WIDTH HALF MAXIMUM DERIVATION

This appendix is provided to justify the Full Width Half Maximum, or FWHM, approach and to characterize its accuracy. The FWHM, or bandwidth technique, is generally applicable to lightly damped poles and has been widely employed in experimental work as far back as 1954. The bandwidth of a resonance peak, or the frequency separation between half power points, is measured from a power spectral density plot, which is equivalent to the square of the displacement frequency response plot. The damping ratio is then computed as

$$\zeta \approx \frac{\Delta\omega}{2\omega_n}, \quad (\text{B.1})$$

where $\Delta\omega$ is the measured bandwidth, and ω_n is the measured undamped natural frequency. The natural frequency is generally approximated to be equal to the damped natural frequency which can be found from the displacement magnitude frequency response as the frequency of its maximum. An assumption of light damping, or $\zeta < 0.1$, is required by this simplification as well as the discarding of higher order ζ terms that was

required to obtain Equation (B.1) in the first place. The reason for this requirement is shown below.

A power spectral density function is equal to the square of the magnitude response function. For a proportionally damped, second order system, the power spectrum of the displacement response to a forcing input is given by

$$P(r) = |H(r)|^2 = \frac{a}{(1-r^2)^2 + 4\zeta^2 r^2}, \quad (\text{B.2})$$

where a is a constant gain that depends upon the particulars of the system being modeled but is assumed not to depend on frequency in the following derivation. For a mass-spring-damper system, a would be equal to the inverse of the square of the mass. The damping ratio is given by ζ . The normalized frequency r represents the ratio of the frequency being considered, ω , to the natural frequency, ω_n , or $r = \frac{\omega}{\omega_n}$. The maximum of the power spectrum occurs at the damped natural frequency which gives $r = \sqrt{1 - \zeta^2}$. The maximum power is then easily found by substitution into Equation (B.2) which gives

$$P_{\max} = \frac{a}{4\zeta^2 - 3\zeta^4}. \quad (\text{B.3})$$

To find the bandwidth, the half power points must first be located. Half of the maximum power given in Equation (B.3) can be equated to the power spectrum to give

$$P_{\text{half max}} = \frac{a}{(1 - r_{\text{half max}}^2)^2 + 4\zeta^2 r_{\text{half max}}^2} = \frac{a}{8\zeta^2 - 6\zeta^4}. \quad (\text{B.4})$$

Solving for $r_{\text{half max}}$ gives

$$r_{\text{half max}} = \sqrt{1 - 2\zeta^2 \pm \zeta\sqrt{8 - 4\zeta^2}}. \quad (\text{B.5})$$

The normalized bandwidth is merely the difference between the two solutions for the half power points given in Equation (B.5). Subtracting these two solutions gives a normalized bandwidth of

$$\Delta r = \sqrt{1 - 2\zeta^2 + \zeta\sqrt{8 - 4\zeta^2}} - \sqrt{1 - 2\zeta^2 - \zeta\sqrt{8 - 4\zeta^2}}. \quad (\text{B.6})$$

Simplification produces

$$\Delta r = \sqrt{2 - 4\zeta^2 - 2\sqrt{1 - 8\zeta^2 + 6\zeta^4}}. \quad (\text{B.7})$$

After squaring Equation (B.7) twice and using the quadratic formula to solve for ζ^2 , the damping ratio solution is given by

$$\zeta = \sqrt{2 - \Delta r^2 - 4\rho - 2\sqrt{1 - 8\rho + 6\rho^2}}, \quad (\text{B.8})$$

where

$$\rho \equiv \frac{1}{16} \left(2\Delta r^2 + 4 \pm \sqrt{6\Delta r^4 + 8\Delta r^2 + 16} \right)^2. \quad (\text{B.9})$$

Examining Equation (B.7) further, the simplifications required to obtain the approximation given in Equation (B.1) become obvious. Consider the $\sqrt{1 - 8\zeta^2 + 6\zeta^4}$ term in this equation. For $\zeta < 0.1$, the addition and subtraction of powers of ζ has little effect on the value of this term and the term can be assumed to be one. For $\zeta = 0.1$, the proposed worst case, this term would equal .9595 and the error in this term produced by assuming it equal to one is 4%. By allowing this assumption the bandwidth formula simplifies to

$$\Delta r = \sqrt{2 - 4\zeta^2 - 2} = 2\zeta. \quad (\text{B.10})$$

Returning to traditional notation from the normalized frequency notation gives

$$\frac{\Delta\omega}{\omega_n} = 2\zeta \quad (\text{B.11})$$

Solving for ζ produces the Equation (B.1). A plot of the percent error between the exact value of the damping ratio given by Equation (B.8) and the approximation given by Equation (B.1) is given in Figure B.1. From the plot it can be seen that utilizing Equation (B.1) introduces less than 1% error for $\zeta < 0.1$. In addition, the error in this approximation decreases logarithmically for decreasing damping ratio.

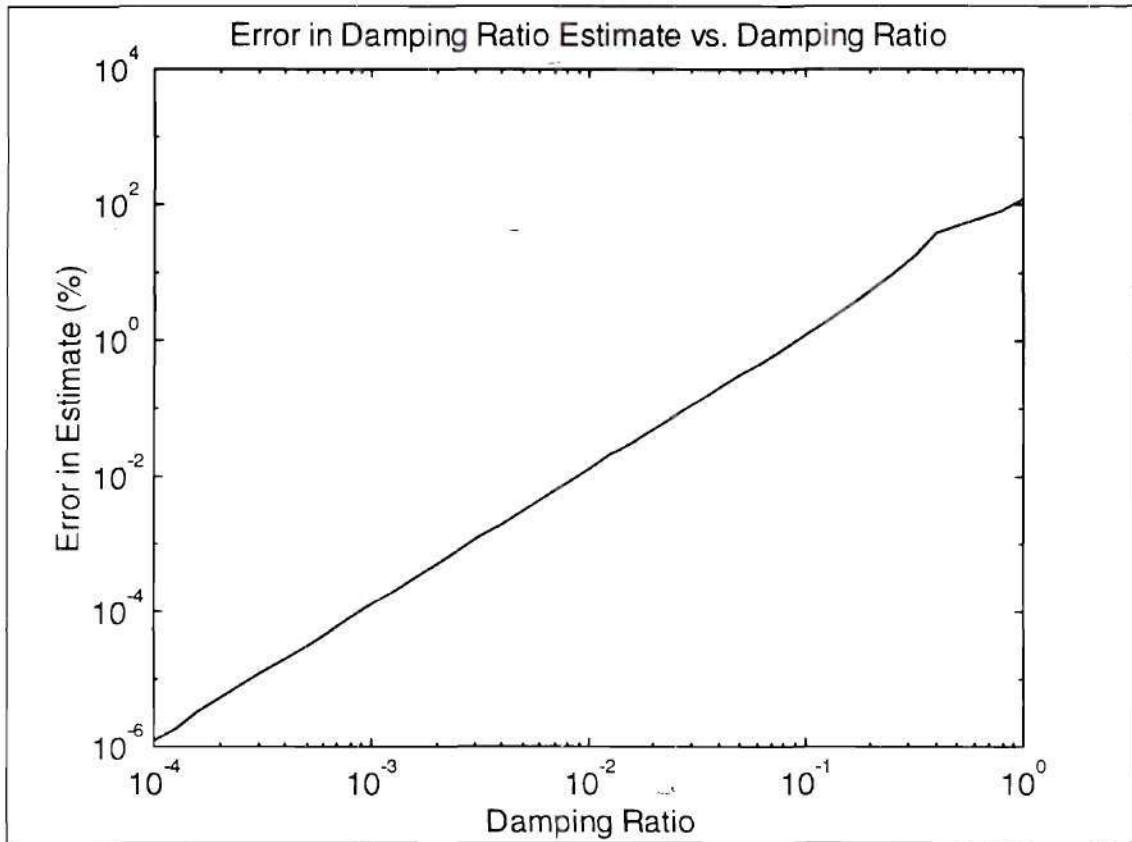


Figure B.1 FWHM Calculation Error

APPENDIX C

VARIABLE RESIDUE METHOD OF POLE PARAMETERIZATION

On method considered for pole parameterization from empirical transfer functions was the Variable Residue Method recently developed by Lin, Lim and Liew in 1995. Because of it's timeliness, a brief summary of the method, and justification for its rejection for on-line system identification is given here. The method relies upon the fact that a Nyquist plot of the frequency response near a pole approximates a circle tangent to and below the real axis. Three adjacent frequency bins denoted by the subscripts a , b , c , in this region around the r^{th} mode are selected and the matrix relationship in Equation (C-12) is solved to produce the estimated complex eigenvalue, λ_r , and the complex gain or modal constant, denoted by C_r .

$$\begin{bmatrix} \Delta_{ba}(\omega_c^2 - \omega_a^2) - \Delta_{ca}(\omega_b^2 - \omega_a^2) & \Delta_{ca}\Delta_{ba}(\omega_c^2 - \omega_b^2) \\ \Delta_{ba}(\omega_c^2 - \omega_b^2) - \Delta_{cb}(\omega_b^2 - \omega_a^2) & \Delta_{cb}\Delta_{ba}(\omega_c^2 - \omega_a^2) \end{bmatrix} \begin{bmatrix} \lambda_r \\ C_r \end{bmatrix} = \begin{bmatrix} \Delta_{ca}\Delta_{ba}\omega_a^2(\omega_c^2 - \omega_b^2) \\ \Delta_{cb}\Delta_{ba}\omega_b^2(\omega_c^2 - \omega_a^2) \end{bmatrix} \quad (\text{C-12})$$

The eigenvalue, λ_r , can then be used to determine the modal frequency and modal damping ratios according to Equations (C-13) and (C-14). Furthermore, multiple

estimates of the modal parameters can be computed and averaged from other triplets of neighboring frequency bins near the targeted mode. It is this averaging, either weighted or unweighted, that improves the robustness of the algorithm over methods that simply search for a local maximum in the data and utilize FWHM for damping ratio estimation.

$$\omega_r \Big|_{\omega=\omega_b} = \sqrt{\text{Re}(\lambda_r)} \quad (\text{C-13})$$

$$\eta_r \Big|_{\omega=\omega_b} = \frac{\text{Im}(\lambda_r)}{\omega_r^2} \quad (\text{C-14})$$

An additional benefit of VRM is the ability to characterize and quantify the nonlinearity of a system by comparing the modal parameters estimated by multiple triplets of the frequency spectra values near a resonant peak. With the VRM algorithm the correlation of the modal parameters with the associated vibration amplitude, the height of the frequency response function at that location, can be easily computed for multiple frequency response function triplets near the resonant peak.

The validity of the VRM algorithm in a low-noise environment was verified in the simulations shown in Figure C.1 through Figure C.3. The frequency response of a single degree of freedom system with modal damping was generated analytically and utilized as input to the VRM algorithm implemented in Matlab. As can be seen in Figure C.2 and Figure C.3, for low noise levels (SNR = 10000), the estimated modal parameters can be

estimated accurately when parameters are computed by averaging within the flat region surrounding the pole under consideration.

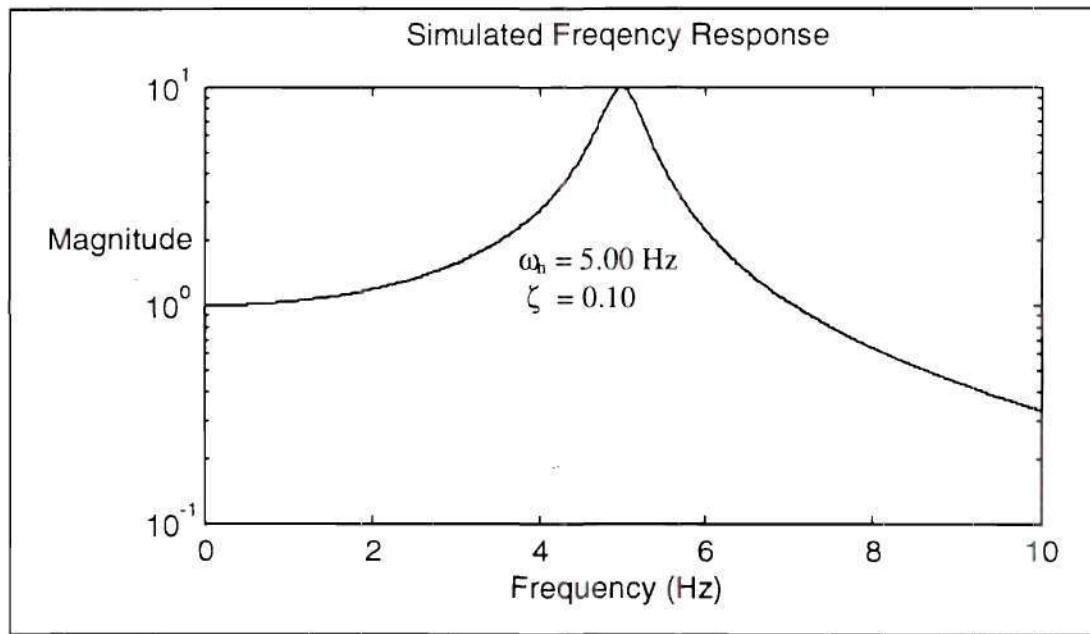


Figure C.1 Empirical Transfer Function Simulation

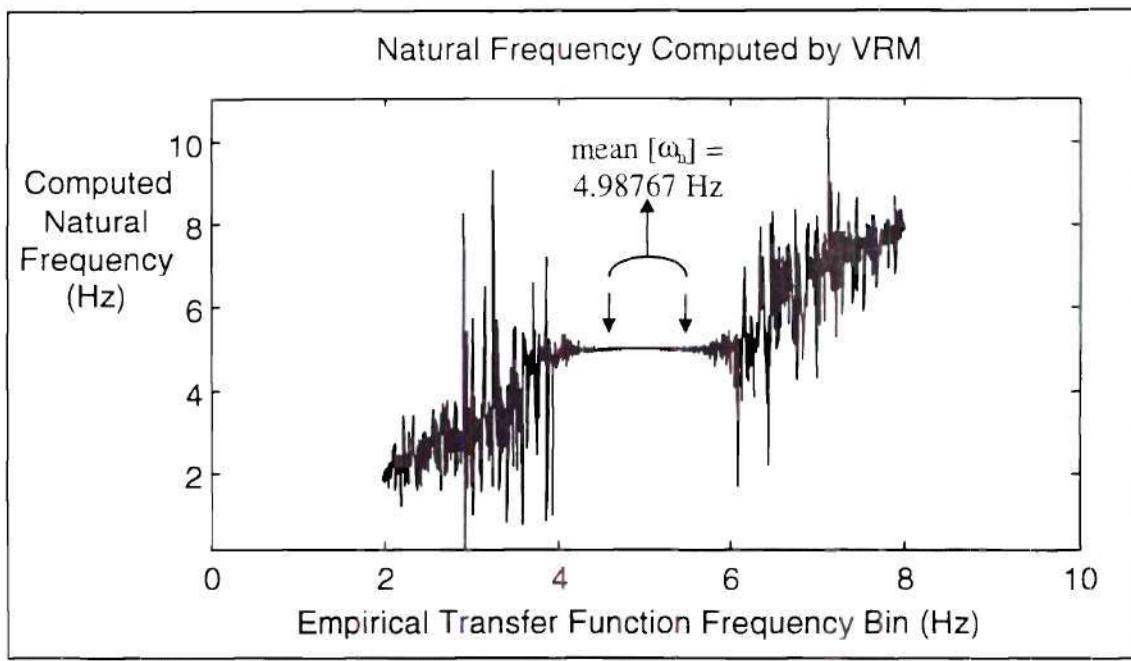


Figure C.2 Natural Frequency Computed From Successive Transfer Function Frequency Bins Using the VRM Algorithm

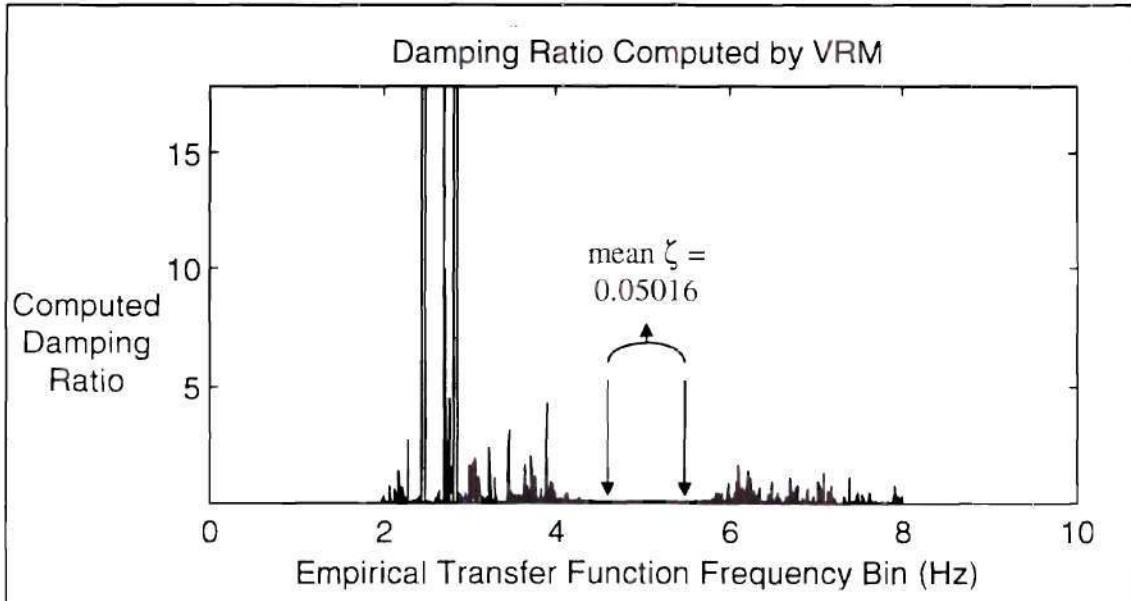


Figure C.3 Damping Ratios Computed From Successive Transfer Function Frequency Bins Using the VRM Algorithm

When more common noise levels of 0.1-10% of the signal were simulated as in Figure C.4, the VRM algorithm was less successful. The assertion by Lin, Lim, and Liew, that the VRM algorithm is practical for experimental application is contradicted by simulation results presented here.

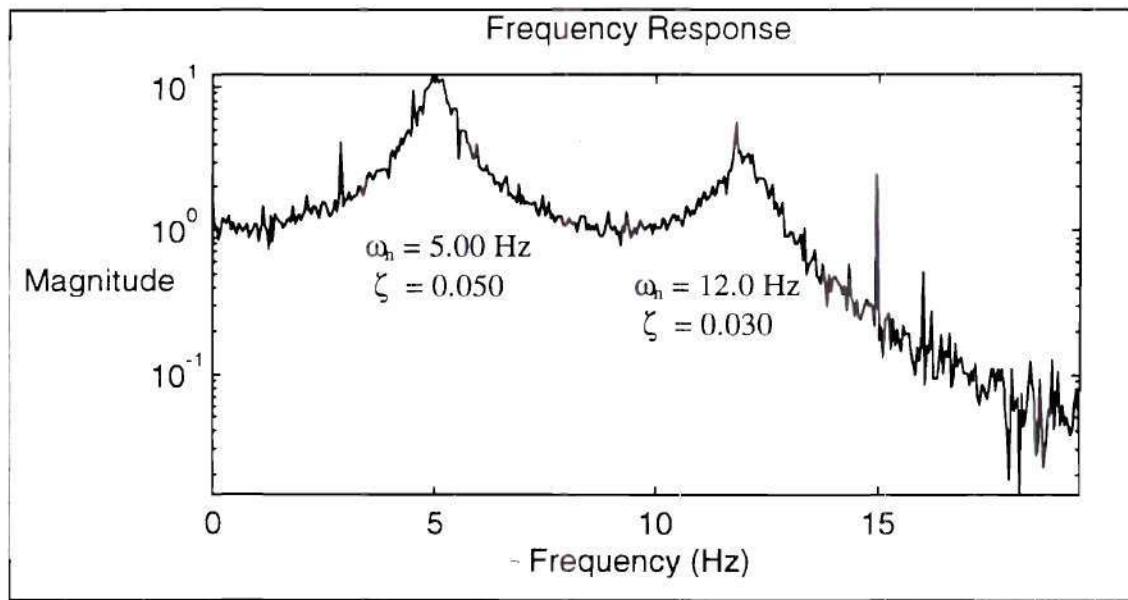


Figure C.4 Simulated Frequency Response for Two Serially Linked Second Order Systems

After considering both the benefits and the limitations of VRM, for experimental work VRM was not utilized to estimate modal parameters in real time. This is due to the fact that in noisy environments, VRM estimates of natural frequency are only as accurate as the initial estimate of the peak. However, utilizing VRM for damping ratio computations did provide an order of magnitude estimate and improved robustness over standard FWHM methods. In Figure C.5 it can be seen that the slope of the natural

frequency estimate graph is not at all flat in the regions near the two poles, and in fact its slope relative to the frequency of the utilized transfer function bins is generally equal to one. However, in Figure C.6, the damping ratio estimates near a pole do show some consistency though they do exhibit a large variance.

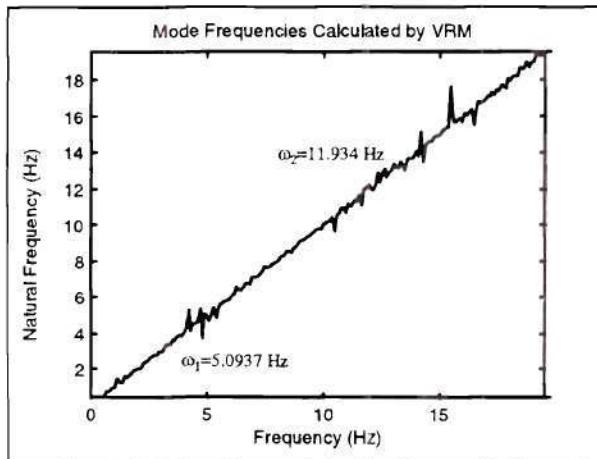


Figure C.5 VRM Calculated Natural Frequency in a Noisy Environment

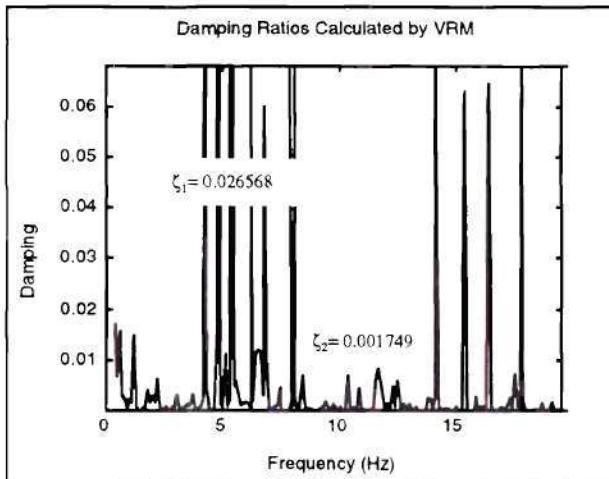


Figure C.6 VRM Calculations of Damping Ratio in a Noisy Environment

APPENDIX D

RECURSIVE LEAST SQUARES

The recursive-least-squares, or RLS, algorithm is extremely useful for both signal estimation and system identification. Employed directly as a system identification algorithm in the time domain, RLS allows the solution to a time evolving matrix equation by utilizing previous solution information recursively to reduce the computational load from $O(N^2)$ for direct solution methods to $O(N)$ for RLS, where N is the order of the linear model employed. Without this reduction in computational requirements, real-time system identification would not be possible for moderately complex linear models. In addition, RLS has found many uses as a signal estimation algorithm. Tzes and Yurkovich used a first order, SISO RLS algorithm to reduce the variance in a transfer function estimate in the "hybrid time-frequency domain."

Because of the usefulness of RLS, and because derivations available in most texts are often unnecessarily complex, application-specific, and incomplete, the general RLS solution is derived here. In addition, its applicability to real-time system identification in the time domain and its specialization to the Tzes and Yurkovich application for frequency domain system identification are explored. A comparison of the computation requirements of these methods is provided. In addition, a generalization of the Tzes and

Yurkovich method to MIMO system models is shown to evolve directly from the general RLS solution.

The recursive-least-squares adaptive filter has coefficients that weight the previous inputs and outputs to produce the current outputs according to

$$\mathbf{d}_k = \mathbf{W}_k \mathbf{x}_k + \mathbf{e}_k \quad (\text{D.1})$$

The vector \mathbf{e}_k contains the Gaussian distributed, uncorrelated, random noise signals at time step k . The vector \mathbf{d}_k lists the desired filter outputs, or, in the system identification role, the observed outputs of a physical plant. For q outputs this gives

$$\mathbf{d}_k = \begin{bmatrix} d_1[k] \\ \vdots \\ d_q[k] \end{bmatrix}. \quad (\text{D.2})$$

When utilizing the RLS algorithm for system identification, q would be equal to the number of measured plant outputs. Likewise, \mathbf{x}_k is a vector of N previous filter inputs for each of the p separate inputs, which are generally the same as the inputs to a physical plant, often denoted $\mathbf{u}[k]$ in control theory. Its structure is defined by

$$\mathbf{x}_k = \begin{bmatrix} x_1[k] \\ \vdots \\ x_p[k] \\ \vdots \\ x_1[k - N + 1] \\ \vdots \\ x_p[k - N + 1] \end{bmatrix} = \begin{bmatrix} \mathbf{u}[k] \\ \vdots \\ \mathbf{u}[k - N + 1] \end{bmatrix}. \quad (\text{D.3})$$

In this formulation the RLS model is a finite impulse response filter. All that is required to implement an infinite impulse response filter is to add additional rows to the subvectors of \mathbf{x}_k to contain the filtered outputs d_1, \dots, d_n that should be fed back into the filter. The resulting input vector would then be

$$\mathbf{x}_k = \begin{bmatrix} x_1[k] \\ \vdots \\ x_m[k] \\ d_1[k] \\ \vdots \\ d_n[k] \\ \vdots \\ x_1[k - N + 1] \\ \vdots \\ x_m[k - N + 1] \\ d_1[k - N + 1] \\ \vdots \\ d_n[k - N + 1] \end{bmatrix}, \quad (\text{D.4})$$

The dimension p is then determined by $p = m + n$. \mathbf{W}_k is the linear system model being identified or the time varying weighting for filtering and signal estimation. The structure of the weighting matrix is given by

$$\mathbf{W}_k = \begin{bmatrix} w_{1,1,0}[k] & \cdots & w_{1,q,0}[k] \\ \vdots & & \vdots \\ w_{p,1,0}[k] & \cdots & w_{q,p,0}[k] \\ \vdots & & \vdots \\ w_{Np-p+1,1,N-1}[k] & \cdots & w_{Np-p+1,q,N-1}[k] \\ \vdots & & \vdots \\ w_{Np,1,N-1}[k] & \cdots & w_{Np,q,N-1}[k] \end{bmatrix}. \quad (\text{D.5})$$

The idea of least-squares estimation is to minimize the error between the desired (or observed) system output and the predicted system output over all of the observed inputs and outputs. The RLS algorithm assumes a square of the norm of the error signal as a cost function. This minimization is formulated as

$$\min_{\mathbf{W}_k} \frac{1}{k+1} \sum_{m=0}^k \left\| \mathbf{d}_m - \mathbf{W}_k^T \mathbf{x}_m \right\|^2. \quad (\text{D.6})$$

However, when the system being identified is non-stationary it is preferable to weight current data more heavily than past data. This can be accomplished by windowing the cost function in time. For exponential windowing the minimization problem becomes

$$\min_{\mathbf{W}_k} \frac{1}{k+1} \sum_{m=0}^k \| \mathbf{d}_m - \mathbf{W}_k^T \mathbf{x}_m \|^2 \alpha^{k-m}, \quad (\text{D.7})$$

where α is a *convergence factor* that implements a trade-off between tracking of a time-varying system and the steady-state error of the model. The *convergence factor*, or as Tzes and Yorkovich refer to it, the *forgetting factor*, must have a value between zero and one to ensure convergence, such that

$$0 < \alpha < 1. \quad (\text{D.8})$$

The solution to this minimization problem can be found by equating the norm of the error to zero over all m and solving this over-determined system of equations by minimizing of the norm of the error with respect to \mathbf{W}_k . Taking its partial derivative with respect to \mathbf{W}_k and setting it equal to zero gives

$$\frac{\partial}{\partial \mathbf{W}_k} \sum_{m=0}^k \| \mathbf{d}_m - \mathbf{W}_k^T \mathbf{x}_m \|^2 \alpha^{k-m} = 0, \quad (\text{D.9})$$

or

$$\sum_{m=0}^k \alpha^{k-m} \frac{\partial}{\partial \mathbf{W}_k} (\mathbf{d}_m - \mathbf{W}_k^T \mathbf{x}_m)^T (\mathbf{d}_m - \mathbf{W}_k^T \mathbf{x}_m) = 0. \quad (\text{D.10})$$

Carrying out the multiplication gives

$$\sum_{m=0}^k \alpha^{k-m} \frac{\partial}{\partial \mathbf{W}_k} (\mathbf{d}_m^\top \mathbf{d}_m - \mathbf{d}_m^\top \mathbf{W}_k^\top \mathbf{x}_m - \mathbf{x}_m^\top \mathbf{W}_k \mathbf{d}_m + \mathbf{x}_m^\top \mathbf{W}_k \mathbf{W}_k^\top \mathbf{x}_m) = 0, \quad (\text{D.11})$$

and evaluating the partial derivative leaves

$$\sum_{m=0}^k \alpha^{k-m} (-2\mathbf{x}_m^\top \mathbf{d}_m + 2\mathbf{W}_k^\top \mathbf{x}_m^\top \mathbf{x}_m) = 0. \quad (\text{D.12})$$

Moving terms containing the unknown \mathbf{W}_k to the left and canceling common factors produces

$$\sum_{m=0}^k \alpha^{k-m} \mathbf{W}_k^\top \mathbf{x}_m^\top \mathbf{x}_m = \sum_{m=0}^k \alpha^{k-m} \mathbf{x}_m^\top \mathbf{d}_m. \quad (\text{D.13})$$

After transposing, this equation becomes

$$\sum_{m=0}^k \alpha^{k-m} \mathbf{x}_m^\top \mathbf{x}_m^\top \mathbf{W}_k = \sum_{m=0}^k \alpha^{k-m} \mathbf{d}_m^\top \mathbf{x}_m. \quad (\text{D.14})$$

Finally, solving for \mathbf{W}_k gives

$$\mathbf{W}_k = \left(\sum_{m=0}^k \alpha^{k-m} \mathbf{x}_m^\top \mathbf{x}_m \right)^{-1} \sum_{m=0}^k \alpha^{k-m} \mathbf{d}_m^\top \mathbf{x}_m \quad (\text{D.15})$$

To produce a concise recursive formula for the solution of the weighting matrix, the covariance matrix and the cross-covariance matrix are defined as

$$\mathbf{P}_k \equiv \sum_{m=0}^k \mathbf{x}_m \mathbf{x}_m^\top \alpha^{k-m}, \quad (\text{D.16})$$

and

$$\mathbf{R}_k \equiv \sum_{m=0}^k \mathbf{x}_m \mathbf{d}_m^T \alpha^{k-m}, \quad (\text{D.17})$$

respectively. Because both the covariance matrix and the cross covariance matrix are sums over the time step index, the recursive relationships for each can be easily derived as

$$\mathbf{P}_k = \alpha \mathbf{P}_{k-1} + \mathbf{x}_k \mathbf{x}_k^T, \quad (\text{D.18})$$

and

$$\mathbf{R}_k = \alpha \mathbf{R}_{k-1} + \mathbf{x}_k \mathbf{d}_k^T. \quad (\text{D.19})$$

As a result, it becomes possible to substitute Equations (D.18) and (D.19) into the minimization solution for \mathbf{W}_k to produce a recursive solution. However, it is also desirable to avoid the inversion calculation of the covariance matrix. The *matrix inversion lemma* can be used to avoid the inversion of the sum in Equation (D.18). Nonetheless, working from first principles, and substituting the definitions from Equation (D.16) and (D.17) into the solution for \mathbf{W}_k in Equation (D.15) gives

$$\mathbf{W}_k = \mathbf{P}_k^{-1} \mathbf{R}_k. \quad (\text{D.20})$$

Utilizing the recursive relationship in Equation (D.19) gives

$$\mathbf{W}_k = \mathbf{P}_k^{-1} (\alpha \mathbf{R}_{k-1} + \mathbf{x}_k \mathbf{d}_k^T). \quad (\text{D.21})$$

Solving Equation (D.20) for \mathbf{R}_{k-1} and substituting the result into Equation (D.21) produces

$$\mathbf{W}_k = \mathbf{P}_k^{-1} (\alpha \mathbf{P}_{k-1} \mathbf{W}_{k-1} + \mathbf{x}_k \mathbf{d}_k^T). \quad (\text{D.22})$$

In order to eliminate \mathbf{P}_k^{-1} term and utilize the previous calculation of the weighting matrix Equation (D.18) can be solved for \mathbf{P}_{k-1} and the result substituted into Equation (D.22) to produce

$$\mathbf{W}_k = \mathbf{P}_k^{-1} \left[\alpha \left(\frac{1}{\alpha} \mathbf{P}_k - \mathbf{x}_k \mathbf{x}_k^T \right) \mathbf{W}_{k-1} + \mathbf{x}_k \mathbf{d}_k^T \right], \quad (\text{D.23})$$

$$\mathbf{W}_k = \mathbf{W}_{k-1} + \mathbf{P}_k^{-1} \mathbf{x}_k (\mathbf{d}_k^T - \mathbf{x}_k^T \mathbf{W}_{k-1}). \quad (\text{D.24})$$

The remaining \mathbf{P}_k^{-1} term is eliminated using Equation (D.18), producing

$$\mathbf{W}_k = \mathbf{W}_{k-1} + (\alpha \mathbf{P}_{k-1} + \mathbf{x}_k \mathbf{x}_k^T)^{-1} \mathbf{x}_k (\mathbf{d}_k^T - \mathbf{x}_k^T \mathbf{W}_{k-1}), \quad (\text{D.25})$$

Distributing the inversion of a sum to produce a sum of inverses, and then factoring out \mathbf{P}_{k-1}^{-1} to produce a scalar denominator gives

$$\mathbf{W}_k = \mathbf{W}_{k-1} + \frac{\mathbf{P}_{k-1}^{-1} \mathbf{x}_k (\mathbf{d}_k^T - \mathbf{x}_k^T \mathbf{W}_{k-1})}{(\alpha + \mathbf{x}_k \mathbf{P}_{k-1}^{-1} \mathbf{x}_k^T)}. \quad (\text{D.26})$$

All that remains is to develop a recursive solution for \mathbf{P}_{k-1}^{-1} . Interestingly, this inverse does not actually exist until the time index k has exceeded the number of columns in the matrix. Nonetheless, because \mathbf{P}_{k-1} is positive definite, the *matrix inversion lemma* can be applied to Equation (D.18). This can also be shown by the algebraic techniques used to obtain Equation (D.26) from Equation (D.25). The recursive solution is then

$$\mathbf{P}_k^{-1} = (\alpha \mathbf{P}_{k-1} + \mathbf{x}_k \mathbf{x}_k^T)^{-1} = \frac{\mathbf{P}_{k-1}^{-1}}{\alpha + \mathbf{x}_k \mathbf{P}_{k-1}^{-1} \mathbf{x}_k^T}. \quad (\text{D.27})$$

Equations (D.26) and (D.27) can be used to implement the RLS algorithm directly. However more sophisticated implementations can reduce the number of operations required and improve the numerical stability of the algorithm by updating the diagonalization or the QR factorization of the inverted matrices rather than the inverses themselves. This is the technique used in Kalman filtering. Without such techniques, the iteration should not continue for more than 100,000 cycles. Compounding numerical inaccuracies will render the computations of the weighting matrix and the covariance matrix invalid beyond this number of recursions.

However, numerical accuracy concerns do not affect SISO, first order filters. For this simplified case, all matrices that must be inverted become scalars. When such an RLS filter is applied to a data set of frequency-domain information evolving in time, the Tzes and Yurkovich solution results. The RLS algorithm, as implemented directly by Equations (D.26) and (D.27) for $p=1$, $q=1$, and $N=1$, can be simplified using scalar algebra to produce

$$w_k = w_{k-1} + \frac{p_{k-1}x_k}{(\alpha + x_k^2 p_{k-1})} (d_k - x_k w_{k-1}), \quad (\text{D.28})$$

$$p_k^{-1} = \frac{p_{k-1}^{-1}}{\alpha + p_{k-1}^{-1} x_k^2}. \quad (\text{D.29})$$

This form is identical to that utilized by Tzes and Yurkovich in published works except for typographical errors in the formulas that were published.

APPENDIX E

ADAPTIVE PIECEWISE POLYNOMIAL REGRESSION

One possible implementation of the frequency domain smoothing suggested by Tzes and Yurkovich is boxcar filter on the frequency domain data, or a first order piecewise polynomial regression. These methods are very successful at reducing the variance in the frequency domain data. However, such a smoothing filter will often flatten desired features such as resonant peaks in the frequency domain resulting in biased damping ration estimates. Figure E.1 shows an example of this effect on a simple time-domain chirp signal. The input for the simulation was a sinusoid of increasing frequency with 10% pseudo-random noise. An improvement was evident when the same signal was processed by an automatic order adjusting piecewise polynomial curve fit. The improved results are shown in Figure E.2. The adaptive smoothing algorithm retains the linear regression's strong rejection of noise in the flat region, while tracking the underlying noise-free signal much more closely at the sharper peaks in the high frequency region. The underlying signal is depicted with a dashed line that does not deviate significantly from the smoothed noise signal except in the flat region. The window width in both cases was fixed at 17 bins, or 8 bins to either side of the point being smoothed. This

corresponds to a $\Delta_i = 8$ for all i in Tzes and Yurkovich's frequency domain smoothing equation.

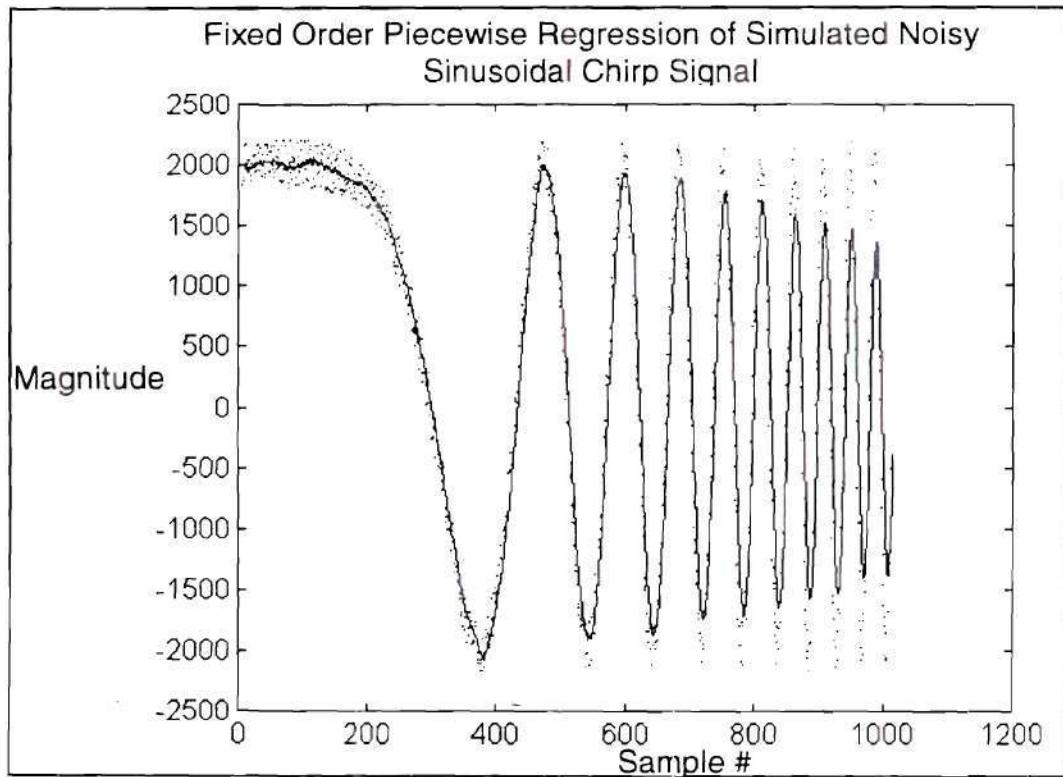


Figure E.1 Example of Peak Flattening Caused by Fixed Order Piecewise Linear Regression

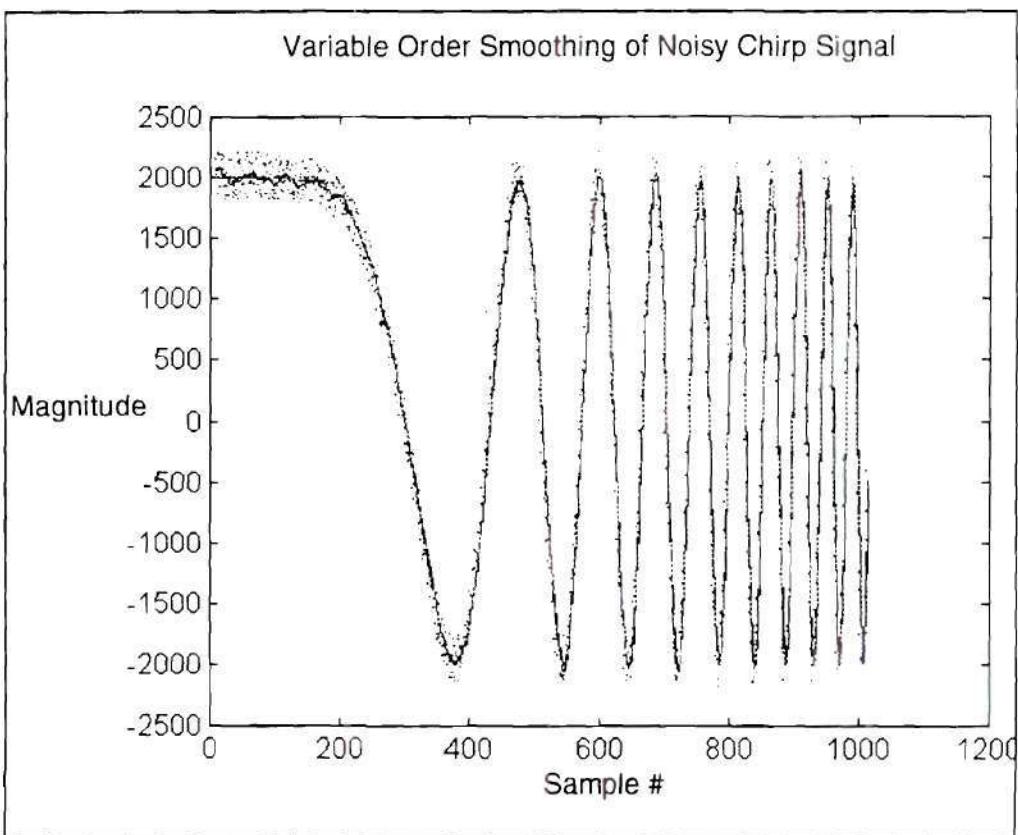


Figure E.2 Variable Order Piecewise Polynomial Regression Example

In spite of its noise rejection benefits, automatic order adjusting piecewise polynomial regression was not used online in experimental work due to its high computational cost. The least squares fitting process produces a system of linear equations must be solved for each frequency bin. In addition, the matrix equations are often badly scaled, if the data within a window varies over a large dynamic range, as is the case near a resonance peak in the magnitude of a frequency spectrum or transfer function. When a fixed order, fixed window width, regression is used, most of the computation can

be performed off-line using Vandermonde matrices, reducing the on-line computations to those required for any arbitrary linear filter.

BIBLIOGRAPHY

1. Bendat, J. S.; Piersol, A. G.; *Engineering Applications of Correlation and Spectral Analysis*, (New York: John Wiley & Sons, Inc., 1980) p. 55.
2. Book, W. J. "Structural Flexibility of Motion Systems in the Space Environment." *IEEE Transactions on Robotics and Automation*, (New York: Institute of Electrical and Electronics Engineers, 1989) Vol. 9, Oct. 1993, p. 524.
3. Cannon, D. W.; Magee, D. P.; Book, W J.; Lew, J. Y.; "Experimental Study on Micro/Macro Manipulator Vibration Control," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, Vol. 3, April, 1996, p. 2549.
4. Cannon, D. W., *Command Generation and Inertial Damping Control of Flexible Macro-Micro Manipulators*, MS Thesis, Georgia Institute of Technology, May, 1996 p. 16.
5. Gardner, W. A.; *Statistical Spectral Analysis, A Nonprobabilistic Theory* (Englewoods Cliffs, New Jersey: Prentice-Hall, Inc., 1988) p. 74-76.
6. Giordano, A. A.; Hsu, F. M.; *Least Square Estimation with Applications to Digital Signal Processing* (New York: John Wiley & Sons, 1985) p. 247-248.
7. Guillaume, P.; Pintelon, R.; Schoukens, J.; "Robust Parametric Transfer Function Estimation Using Complex Logarithmic Frequency Response Data," *IEEE Transactions on Automatic Control*, Vol 40, No. 1, July, 1995, p. 1181.
8. Khorrami, F.; Jain, S.; Tzes, A.; "Experimental Results on Adaptive Nonlinear Control and Input Preshaping for Multi-link Flexible Manipulators," *Automatica*, Vol. 31 No1, 1995, pp. 83-97.
9. Khorrami, F.; Jain, S.; Tzes, A.; "Experiments on Rigid Body-Based Controllers with Input Preshaping for a Two-Link Flexible Manipulator," *IEEE Transactions on Robotics and Automation*, Vol. 10, Feb. 1994, p. 55.
10. Lin, R. M.; Lim, M. K.; Liew, K. M.; "Variable Residue Method for Modal Parameter Estimation," *Transactions of the ASME*, Vol. 117, Oct. 1995, p. 392.

11. Magee, D. P., *Optimal Arbitrary Time-Delay Filtering to Minimize Vibration in Elastic Manipulator Systems*, Ph.D. Thesis, Georgia Institute of Technology, 1996, p. 88, 161-207.
12. Magee, D. P.; Book, W. J.; "Filtering Micro-Manipulator Wrist Commands to Prevent Flexible Base Motion," *Proceedings of the American Control Conference*, June 1995, pp.927-928.
13. Milford, R. I.; Asokanthan, S. F.; "Experimental On-Line Frequency Domain Identification and Adaptive Control of a Flexible Slewing Beam," *Journal of Dynamic Systems, Measurement and Control*, Vol. 118, March 1996, p. 58-64.
14. Mitra, S. K.; Kaiser, J. F.; *Handbook for Digital Signal Processing*, (New York: John Wiley & Sons Inc, 1993) p1117, p1179.
15. Nashif, A. D; Jones, D. I. G.; Henderson, John P., *Vibration Damping*, (New York: John Wiley & Sons, Inc., 1985), p. 170.
16. Newland, D. E., *Mechanical Vibration Analysis and Computation*, (New York: John Wiley and Sons, Inc. 1989) p. 290.
17. Oppenheim, A. V.; Schafer, R. W.; *Discrete-Time Signal Processing* (Englewood Cliffs, New Jersey: Prentice Hall, 1989, p. 717.
18. Poor, H. V., *An Introduction to Signal Detection and Estimation*, (New York: Dowden and Culver, 1994) p.230.
19. Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P.; *Numerical Recipes in C*, (New York: Cambridge University Press, 1994) p512-513.
20. Safak, E.; Celebi, M.; "Recorded Seismic Response of Pacific Park Plaza," *Journal of Structural Engineering*, Vol. 118, June, 1992, p. 1566,
21. Schiff, A.. J., "Identification of Large Structures Using Data from Ambient and Low Level Excitations," *System Identification of Vibrating Structures, Mathematical Models from Test Data*, (New York: The American Society of Mechanical Engineers, 1972), p. 110, 106.
22. Sciacicco, L.; Siciliano, B.; *Modeling and Control of Robot Manipulators*, (New York: The McGraw-Hill Companies, Inc., 1996). p.129.

23. Seering, W.; Singhouse, W.; Singer, N.; "Residual Vibration Reduction Using Vector Diagrams to Generate Shaped Inputs," *Journal of Mechanical Design*, (New York: American Society of Mechanical Engineers, 1990) Vol. 116, No. 1, June, 1994 , p. 654.
24. Singer, N.; S., W.; "Controlling Vibration in Remote Manipulators." *ASME Journal of Dynamic Systems, Measurement and Control*, March 1988, p. 1.
25. Strang, G., *Introduction to Applied Mathematics*, (MA: Wellesley-Cambridge Press, 1986).
26. Szidarovszky, F.; Bahill, A. T.; *Linear Systems Theory*, (Boca Raton, FL: CRC Press, 1991).
27. Tzes, A. P.; Yurkovich, S.; "Application and Comparison of On-Line Identification Methods for Flexible Manipulator Control," *The International Journal of Robotics Research*, Vol. 10, No. 5, October, 1991, p. 515-518.
28. Tzes, A. P.; Yurkovich, S.; "An Adaptive Input Shaping Control Scheme for Vibration Suppression in Slewing Flexible Structures," *IEEE Transactions on Control Systems Technology*, Vol 1, June, 1993, p. 114.
29. Tzes, A.; Yurkovich, S.; "A Frequency Domain Identification Scheme for Flexible Structure Control," *Transactions of ASME: Journal of Dynamics Measurement and Control*, Vol. 112, Sept. 1990, p. 429.
30. Williams, O. A., *System Impulse Response from Digital Cross Correlation of Additive Input Noise and Output Signal*, MS Thesis, Georgia Institute of Technology, 1969. Ch. 3, p. 15.
31. Yang, T.; Yang, J. C. S.; Kudva, P.; "Load-adaptive Control of a Single-link Flexible Manipulator," *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 22., Jan-Feb 1992, p85