REAL TIME IDENTIFICATION OF LARGE SPACE STRUCTURES

by

JANICE ELAINE VOSS

B.S., Purdue University (1975)
S.M., Massachusetts Institute of Technology (1977)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1987

Signature of Author Signature redacted
Department of Aeronautics and Astronautics
December 21, 1986

Certified by Signature redacted
Professor Wallace E. VanderVelde
Thesis Supervisor

Certified by Signature redacted
Professor Bruce K. Walker
Thesis Supervisor

Certified by Signature redacted
Dr. Philip D. Hattis
Thesis Supervisor

Accepted by Signature redacted
Professor Harold Y. Wachman
Chairman, Departmental Graduate Committee

Page 71

REAL TIME IDENTIFICATION OF LARGE SPACE STRUCTURES

by

JANICE ELAINE VOSS

Submitted to the Department of Aeronautics and Astronautics on January 16, 1987 in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Aeronautics and Astronautics.

ABSTRACT

This thesis examines identification of frequencies, damping ratios, and mode shapes of large space structures (LSSs) in real time. Real time processing allows for quick updates of modal processing after a reconfiguration or structural failure.

Recursive lattice least squares (RLLS) was selected as the baseline algorithm for the identification. Simulation results on a one dimensional 'LSS' demonstrated that it provided good estimates, was not ill-conditioned in the presence of under-excited modes, allowed activity by a supervisory control system which prevented damage to the LSS or excessive drift, and was capable of real-time processing for typical LSS models.

A suboptimal version of RLLS, which is equivalent to simulated parallel processing, was derived. Applying it to the same data as the optimal version showed that the number of computations per step could be reduced by a factor of four with little accuracy penalty.

The possibility of improving these results by using an input optimized for the LSS was examined. A method for constructing optimal inputs was presented and applied to simple LSSs. The identification results using an optimal input were compared to those using a pseudo-random binary sequence as the input. The optimal input significantly improved the accuracy of the estimates.

A NASTRAN model of the dual keel U.S. space station was used to demonstrate the input/identification algorithm package in a more realistic simulation. Because the first eight flexible modes were very close together, the identification was much more difficult than in the simple examples. Even so, the model was accurately identified in real time.

Thesis supervisor: Wallace E. VanderVelde
Title: Professor of Aeronautics and Astronautics

Thesis supervisor: Bruce K. Walker
Title: Professor of Aeronautics and Astronautics

Thesis supervisor: Philip D. Hattis
Title: Section Chief, Advanced Programs

Acknowledgements

The following people share in the credit for this thesis:

My parents, Louise Voss and Dr. James Voss, for their support, encouragement, and advice since the day I first described a C clamp as a large micrometer;

My husband, (soon-to-be-Dr.) Russell Howard, for helping me stay sane and providing a break from the simulation world of my thesis by providing cars that frequently needed real repairs;

Dr. William Widnall, whose help through the early stages of my thesis was invaluable in keeping me on track;

The members of my committee, who not only provided excellent advice and comments, but who were willing to accept such a dangerous assignment (during my graduate career, two committee members have died, one took a leave of absence without attending even one committee meeting and never came back, and two left for other jobs; there was one good story--one was promoted off my committee);

The members of the Space Systems Laboratory, whose enthusiam helped get me through the slow times;

Dr. Richard Battin, who straightened out M.I.T.'s bureaucracy for me and helped me get a position at Draper Laboratory;

Dr. Steve Croopnick, who offered me the position which led to this thesis.

Table of Contents

	Page
Abstract	2
Acknowledgements	3
Table of Abbreviations, Acronyms, and Terminology	6
Table of Symbols and Notational Conventions	6
Chapter 1: Introduction	14
1.1 The Large Space Structure (LSS) Problem	14
In Orbit Identification	15
Closely Spaced Frequencies	15
1.2 Thesis Goal	15
Chapter 2: Identification Algorithms	17
2.1 Choosing a Model	17
Parameterizations: SISO	18
Parameterizations: MIMO	20
Modal Forms	23
2.2 Choosing an Algorithm	27
Recursive Least Squares	30
Recursive Maximum Likelihood	31
Recursive Extended Least Squares	32
Recursive Generalized Least Squares	32
2.3 Inputs	33
2.4 LSS Algorithm (Recursive Lattice Least Squares or RLLS)	33
Derivation	34
Extension of Derivation	35
Correlated Noise	40
Evaluating Performance	40
Chapter 3: Computational Considerations	53
3.1 Computational Characteristics of RLLS	53
Cycle Time	53
Recursive-by-Order Property	59
Parallel Processing	64

Page

3.2 Suboptimal, Low Cycle Time Recursive Lattice Least Squares 66
No Noise Case 66
Noise Case. 68

Chapter 4: Inputs to Enhance Identification. 76
4.1 Introduction. 76
4.2 Figure of Merit 76
4.3 Necessary (First Order) Conditions. 78
4.4 Sufficient (Second Order) Conditions. 83
4.5 Simple Example. 84
4.6 Solutions for a Simple LSS. 89
Characteristics of the Solution 90
Optimizing with Respect to a Single Parameter 94
Optimizing with Respect to Many Parameters. 101
Limiting State Amplitude. 107
Separating Closely Spaced Frequencies 115
4.7 Comparison with PRBS. 120
4.8 Unknown Elements in Observation Matrix. 129
4.9 MIMO LSS. 130
4.10 Programming Considerations. 133

Chapter 5: Results for a Typical LSS 135
5.1 Dual Keel Space Station Model 135
Model Size. 138
5.2 Simulation Environment. 139
Noise Sources 139
Sensors and Actuators 139
5.3 Results 142

Chapter 6: Conclusions and Recommended Future Work 163

Appendix A: Derivation of Recursive Lattice Least Squares. 166
A.1 Derivation of RLLS as an Orthogonal Projection. 166
A.2 Derivation of RLLS as a Coordinate Transformation 173
A.3 Extension of Derivation 190
Appendix B: Modal Parameterization 3 191
References. 194

Table of Abbreviations, Acronyms, and Terminology

ARMA = auto-regressive, moving average

ARMAX = auto-regressive, moving average, exogenous input

CPU = central processing unit

identifier order = number of stages in identifier = M

LLS = lattice (or ladder) least squares

LS = least squares

LSS = large space structure

MIMO = multiple input, multiple output

PRBS = pseudo-random binary sequence

RELS = recursive extended least squares

RGLS = recursive generalized least squares

RML = recursive maximum likelihood

RLELS = recursive lattice (or ladder) extended least squares

RLLS = recursive lattice (or ladder) least squares

RLS = recursive least squares

SISO = single input, single output

slinky = masses connected by linear springs and viscous dampers (see Figure 1.2.1)

SNR = signal to noise ratio

system order = number of modes in the system model = N/2 = n

UDU^T = way to factor a covariance matrix; U is upper triangular and D is diagonal

Table of Symbols and Notational Conventions

$(\underline{\ })$ = ( ) is a column vector

$(\underline{\ })_R$ = ( ) is a row vector

$(\ )^T$ = transpose of ( )

$(\dot{})$ = derivative of ( )

$(\hat{})$ = estimate of ( )

$\tilde{(\ )}$ = (1) corrected value of ( ) (see RML algorithm in Section 2.2)
(2) ( ) transformed to RLLS form (see equation A12)

$\bar{(\ )}$ = $(\underline{\ })$ calculated with estimated parameters one step less current than those used to get $(\underline{\ })$ (compare equations A21 and A42)

$(\ )^*$ = (1) complex conjugate (see section 2.1)
(2) variables involved in the unknowns estimated by RLLS which are not of direct interest in this thesis i.e. unknowns which describe an ARMAX model in terms of future inputs and outputs (compare equations A23 and A14)

$(\ )^{(n)}$ = the version of ( ) appropriate to an nth order identifier (see equation 2.4.1); similar to $(\ )_n$ except this is used when the orders are not independent so that ( ) depends on the actual order used

$(\ )_i^{(n)}$ = the ith order part of ( ), which was identified with an nth order identifier (see equation 2.4.1)

$(\ )_i$ = (1) partial derivative of ( ) with respect to i (see equation 4.2.1)
(2) order index i.e. minimum order of identifier which would use ( ) (see equation A14); similar to $(\ )^{(i)}$ except this is used when the orders are independent so that ( ) doesn't change with the actual order used

$|(\ )|$ = magnitude of ( ) i.e $(\ )^2$ or $(\underline{\ })^T\ (\underline{\ })$

dim( ) = dimension of ( )

Im( ) = imaginary part of ( )

Re( ) = real part of ( )

α = (1) dimension of many vectors and matrices used in an RLLS identifier = m + p for RLLS = m + 2p for RLELS
(2) parameter targeted by an optimal input (see equation 4.2.1)

β = (1) scalar weighting factor used in RLLS = γ/λ (see equations A31-A33 and Figure 2.4.1)
(2) eigenvalues -- of the matrix F' (see equation 4.3.18) or the matrix F (see section 2.1, "Modal Forms")

$\underline{\beta}$ = vector of unknown parameters which multiplies the past inputs in MIMO Parameterization 3 (see section 2.1, "Parameterizations: MIMO")

γ = (1) scalar weighting factor used in RLLS = $\lambda\ \beta$ (see equation A61 and Figure 2.4.2)

(2) unknown parameters which are part of the vector $\underline{\beta}$ in MIMO Parameterization 3 (see section 2.1, "Parameterizations: MIMO")

$\underline{\gamma}$ = vector containing constants chosen to meet boundary conditions used to calculate optimal inputs (see equation 4.3.18); dim($\underline{\gamma}$) = 2N*(r+1) = 4n*(r+1)

Γ = Hamiltonian (see equation 4.3.14)

$\underline{\varepsilon}$ = error estimates used in RLELS (see "Correlated Noise" in section 2.4); dim($\underline{\varepsilon}$) = p

ζ = modal damping ratio

θ = matrix of unknown parameters to be estimated (see equation 2.2.1)

λ = forgetting factor (see Chapter 2)

$\underline{\lambda}$ = LaGrange multiplier (see equation 4.3.14); dim($\underline{\lambda}$) = N*(r+1)

ν = (1) structural index = number of rows in the observability matrix corresponding to each input (see "Parameterizations: MIMO" in section 2.1)

(2) boundary condition on LaGrange multipliers $\underline{\lambda}$ (see equation 4.3.14)

ν_i = number of rows in the observability matrix corresponding to input_i (see "Parameterizations: MIMO" in section 2.1)

ν_{ij} = dimension of the ijth block of the A matrix in observable canonical form (see "Parameterizations: MIMO" in section 2.1)

$\underline{\pi}$ = $[\ 0\ \cdot\ \cdot\ \cdot\ 0\ 1\ ]^T$ (see equation A4)

$\underline{\phi}$ = (1) vector describing modal shape (see Chapter 1); dim($\underline{\phi}$) = N

(2) vector of regressors, whose elements depend on the identification algorithm being used (see equation 2.2.1 and Appendix A)

(3) eigenvectors of matrix F' (see equation 4.3.18); dim($\underline{\phi}$) = 4n*(r+1) = 2N*(r+1)

Φ = matrix whose columns are the eigenvectors of matrix F' (see equation 4.3.18); size = 4n*(r+1) x 4n*(r+1)

Φ_c = matrix Φ with the top half evaluated at time t=0 and the bottom half evaluated at time t=T (see text after equation 4.3.18); size = 4n*(r+1) x 4n*(r+1)

$\Phi_{R\ (or\ L)}$ = matrix of right (or left) eigenvectors (see section 2.1, "Modal Forms")

ω = modal frequency

ω_c = average or center frequency of two modes, used when optimizing with respect to the frequency difference of the modes (see "Separating Closely Spaced Frequencies" in section 4.6)

$\underline{\xi}$ = augmented state vector used for calculating optimal input (see equation 4.3.5); $\dim(\underline{\xi}) = N*(1+r)$

$\underline{\psi}_i$ = derivative of the estimate of $output_i$ with respect to its unknowns (see equation 2.2.2); $\dim(\underline{\psi}_i)$ = number of columns in θ

$\underline{a}_{ij}$ = vector of unknown parameters which multiplies the past outputs in MIMO Parameterization 3 (see section 2.1, "Parameterizations: MIMO").

A = (1) system matrix as in $\dot{\underline{x}} = A\ \underline{x}$ (elements of A are scalar constants) (see Chapter 3); size = N x N

(2) system matrix as in $\underline{x}_{k+1} = A\ \underline{x}_k$ (elements of A are scalar constants) (see Section 2.1); size = N x N

(3) ARMA system matrix as in $\underline{y} = A\ \underline{y}$ (elements of A are polynomials in inverse powers of the q transform variable) (see section 2.2); size = p x p

b_i = elements of a SISO system control influence matrix (see equation 4.6.2)

$b_{i,j}$ = ijth element of modal control influence matrix (see equation 4.3.1)

B = (1) control influence matrix as in $\dot{\underline{x}} = A\ \underline{x} + B\ \underline{u}$ (elements of B are scalar constants) (see Chapter 3); size = N x m

(2) control influence matrix as in $\underline{x}_{k+1} = A\ \underline{x}_k + B\ \underline{u}_k$ (elements of B are scalar constants) (see Section 2.1); size = N x m

(3) ARMA system matrix as in $\underline{y} = A\ \underline{y} + B\ \underline{u}$ (elements of B are polynomials in inverse powers of the q transform variable) (see section 2.2); size = p x m

c_i = (1) viscous damper constants (see Figures 1.2.1 and 4.6.1)

(2) elements of a SISO system measurement matrix (see equation 4.6.2)

$c_{i,j}$ = ijth element of measurement matrix (see equation 4.3.2)

C = (1) measurement matrix as in $\underline{y} = C\underline{x}$ (see Section 2.1 and Chapter 4); size = p x N

(2) noise influence numerator matrix as in $\underline{y} = A\underline{y} + B\underline{u} + C\underline{e}$ (elements of C are polynomials in inverse powers of the q transform variable) (see section 2.2); size = p x p

(3) damping matrix in finite element model of a system (see section 2.1, "Physical System Model"); size = n x n

D = (1) control measurement matrix as in $\underline{y} = C\underline{x} + D\underline{u}$ (see Section 2.1 and Chapter 4); size = p x m

(2) noise influence denominator matrix as in $\underline{y} = A\underline{y} + B\underline{u} + D^{-1}\underline{e}$ (elements of D are polynomials in inverse powers of the q transform variable) (see section 2.2); size = p x p

(3) diagonal matrix obtained from factoring a covariance matrix as in UDU^T (see below equation A16 and section 2.4)

e = (1) noise on outputs (see Chapter 2); dim($\underline{e}$) = p

(2) vector of regressors (errors) used in the RLLS identifier (see Figures 2.4.1 and 2.4.2, and equations A1 and A21); dim($\underline{e}$) = α

E = constraint on integrated amount of input allowed for the optimal input = $\frac{1}{2}\int_0^T \underline{u}^T \underline{u}\, dt$ (see equation 4.2.3)

F' = augmented system matrix for $\underline{\xi}$ and $\underline{\lambda}$ (see equation 4.3.15); size = 2N*(r+1) x 2N*(r+1)

F = (1) matrix used in commonly used version of the RLLS identifier which approximately equals $E\{\underline{r}\,\underline{e}^T\}$ (see Figure 2.4.1); size = α x α

(2) system matrix of augmented system used for calculating an optimal input i.e. $\dot{\underline{\xi}} = F\,\underline{\xi} + G\,\underline{u}$ (see equation 4.3.6); size = N*(r+1) x N*(r+1)

(3) system matrix ($\dot{\underline{x}} = F\,\underline{x} + G\,\underline{u}$) in system from which a modal form is derived (see section 2.1, "Modal Forms")

G = (1) control influence matrix of augmented system used for calculating an optimal input i.e. $\dot{\underline{\xi}} = F\,\underline{\xi} + G\,\underline{u}$ (see equation 4.3.6); size = N*(r+1) x m

(2) control influence matrix ($\dot{\underline{x}} = F\,\underline{x} + G\,\underline{u}$) in system from which a modal form is derived (see section 2.1, "Modal Forms")

$G^{(n)}$ = matrix of unknown parameters to be estimated (see equation A10). Similar to θ, except that θ is used in a model which predicts present outputs from the past; G is used to predict both present inputs and outputs from the past. Size = α x $n\alpha$

$G_i^{(n)}$ = columns of $G^{(n)}$ from $1+(i-1)*\alpha$ to $i*\alpha$ (see equation A1)

H = (1) measurement matrix of augmented system used for calculating an optimal input i.e. $[\underline{y}_{\alpha_i}] = H\ \underline{\xi}$ (see equation 4.3.7); size = r*p x N*(r+1)

(2) measurement matrix ($\underline{y} = H\ \underline{x} + D\ \underline{u}$) in system from which a modal form is derived (see section 2.1, "Modal Forms")

$H^{(n)}$ = matrix of unknown parameters to be estimated (see equation 2.4.3). Similar to θ, except that θ is used in a model which predicts present outputs from the past; H is used to predict the past (both outputs and inputs) from the present. Size = α x $n\alpha$

I_p = p x p identity matrix

J = cost function for calculating optimal inputs (see equation 4.2.1)

$\underline{k}^{(n)}$ = Kalman gain vector in RLS estimator for unknowns in G matrix (see equation A39); $\dim(\underline{k}^{(n)}) = n*\alpha$

$\underline{k}_n$ or $\underline{k}_n^*$ = Kalman gain vector in RLLS estimator (see equations 2.2.2, 2.4.4, and 2.4.5); $\dim(\underline{k})$ = number of columns in θ (equation 2.2.2) or = α (equations 2.4.4 and 2.4.5)

k_i = (1) weighting factor used in J, the cost function for calculating optimal inputs (see equation 4.2.1)

(2) spring constants (see Figures 1.2.1 and 4.6.1)

K = (1) weighting factors k_i arranged in a matrix (see equation 4.3.8); size = r*p x r*p

(2) stiffness matrix in finite element model of a system (see section 2.1, "Physical System Model"); size = n x n

K_n or K_n^* = unknowns being estimated in RLLS identifier (see Figures 2.4.1 and 2.4.2 and equations A14 and A23); size = α x α

$\underline{l}^{(n)}$ = Kalman gain vector in RLS estimator for unknowns in H matrix (see equation A50); $\dim(\underline{l}^{(n)}) = n*\alpha$

m = number of inputs = $\dim(\underline{u})$

m_i = mass in a slinky (see Figures 1.2.1 and 4.6.1)

M = (1) maximum identifier order = number of stages in identifier (see Figures 2.4.1 and 2.4.2)

(2) transformation matrix used to get from MIMO Parameterization 2 to MIMO Parameterization 3 (see section 2.1)

(3) mass matrix in finite element model of a system (see section 2.1, "Physical System Model"); size = n x n

n = (1) index most commonly used to count identifier order (see Figures 2.4.1 and 2.4.2)

(2) number of modes in a system model (see equation 4.3.1) = N/2

N = (1) dim(stage vector $\underline{x}$) (section 2.1) = twice the system order = 2n

(2) abbreviation for a newton of force

p = number of outputs = dem($\underline{y}$)

P( ) = projection operator which projects on the space spanned by the rows of matrix ( ) (see equation A4)

P_n or P_n^* = covariance matrix, size = (number of columns in θ) x (number of columns in θ) (equation 2.2.3) or α x α (eqs. 2.4.4 and 2.4.5)

q = (1) discrete time transform parameter i.e. q $\underline{y}(k) = \underline{y}(k+1)$

(2) weighting parameter on the input in an alternate form of the cost function for the optimal input (see "Characteristics of the Solution" in section 4.6)

$Q^{(n)}$ = inverse of covariance matrix for estimating $H^{(n)}$ (see equation A49); size = $n*\alpha$ x $n*\alpha$

r = number of parameters being optimized over (see equation 4.3.5)

$\underline{r}_n$ = vector of residuals used in the RLLS identifier (see Figures 2.4.1 and 2.4.2, and equations 2.4.3 and A5); dim($\underline{r}$) = α

$R^{(n)}$ = inverse of covariance matrix P used in the RLS identifier (see equation A38); size = $n*\alpha$ x $n*\alpha$

R_n or R_n^* = inverse of the covariance matrix P_n or P_n^*

S = (1) matrix used to prove sufficient second order conditions for the optimality of the inputs (see equation 4.4.3)

(2) generic matrix which equals $X_{N,t}$ (see eq. A6 and Table A.1)

t = time index, may be either continuous or discrete

t' = time shifted by n units = t - n (see equation A33)

T = duration of an identification run (Chapter 4)

$T^{(n)}$ = lower triangular transformation matrix with ones on the diagonal which converts from RLS to RLLS form (see equation A12)

$\underline{u}$ = vector of inputs; dim($\underline{u}$) = m

U = upper triangular matrix obtained from factoring a covariance matrix as in UDU^T (see below equation A16 and section 2.4)

$\underline{v}$ = estimate of the value of the noise $\underline{e}$ (see RML(II) algorithm in Section 2.2); dim($\underline{v}$) = p

V = (1) variance of output noise (see equation 2.2.2 and Figure 2.4.2)
(2) generic vector or matrix (see equation A6 or Table A.1)

$\underline{w}$ = estimate of the value of the noise $\underline{e}$ (see RML(I) algorithm in Section 2.2); dim($\underline{w}$) = p

W = (1) weighting matrix which penalizes state amplitude in calculating an optimal input (see equation 4.6.4); size = N x N
(2) generic vector or matrix (see equation A6 or Table A.1)

$\underline{x}$ = (1) state vector (section 2.1 and Chapter 4)); dim($\underline{x}$) = N
(2) vector containing regressors = $[\underline{y}^T \ \underline{u}^T]^T$ for least squares algorithms (dim($\underline{x}$) = m+p) = $[\underline{y}^T \ \underline{u}^T \ \underline{e}^T]^T$ for extended least squares algorithms (dim($\underline{x}$) = m+2p) (section 2.4)

$\underline{x}_t$ = $[\ \underline{x}(0) \cdot \cdot \cdot \underline{x}(t)\ ]^T$; size = (t+1) x α (see equation A2)

$\underline{x}_t^{N+1}$ = $\underline{x}_t$ shifted N times with 0 in the vacated elements (see eq. A5)

$X_{N,t}$ = matrix of past inputs and outputs (see equation A3); size = N*α x (t+1)

$\underline{y}$ = vector of outputs; dim($\underline{y}$) = p

$\underline{y}'$ = $\underline{y}$ with the ith element removed (see MIMO Parameterization 3)

Y = generic vector or matrix (see equation A6 or Table A.1)

Chapter 1: Introduction

For the past several years, design and construction of large space structures (LSSs) has been a growing area of interest. These systems will range in size from tens of meters to several hundred meters in their largest dimension. Examples currently of interest are the Langley Mast flight experiment, the Hubble Space Telescope, and the U.S. space station.

1.1 The Large Space Structure Problem

Since launch costs are a major portion of total space system costs, achieving high structural stiffness by structural means will be very costly. This forces space structures to be light and flexible relative to earthbound structures. The low natural frequencies, caused by the low stiffness coupled with the low natural damping characteristic of light metal and composite structures, lead to large response to low frequency disturbances. Space structure control systems must be designed using dynamic models that include some flexible modes. This requires accurate values for the structural frequencies (ω_i), mode shapes (ϕ_i), and modal damping (ζ_i).

These modal parameters can be estimated by analysis of finite element models of the LSS. The accuracy of these estimates is limited by having to approximate a distributed structure with a finite element model, and discrepancies between the expected structure and the actual structure (e.g. use "typical" material property values rather than measuring the exact values for each piece of the structure). The best estimates will be 10%-20% off, and even that accuracy will be achieved only for the lowest frequency modes (ref. 25).

For more accurate estimates, testing is done on a prototype of the space structure. The structure is mounted on a test platform, shaken, and the response is measured. LSSs are too big for ground-based testing of a full-scale prototype, so the final testing will have to be done in orbit.

In Orbit Identification

Doing the identification in orbit creates special problems. Getting equipment and personnel to orbit is expensive (currently $2000/pound)[26], so that testing will be done on flight hardware, not a prototype. Actuators and sensors will be limited to those which will be on the operational system. This limits the number and type of inputs and outputs available. It also limits the inputs to those which don't overstress the structure and don't drive the structure to a state from which the control system can't recover.

There are two options for processing the data - onboard or on the ground. Air-to-ground communications have a limited data transmission capability, so onboard processing is preferred. Autonomy is a goal for the space station, again pointing towards onboard processing. This will limit computing power, CPU time available to devote to identification, and data storage capacity. Due to data storage limitations and for applications involving reconfiguration (like docking a shuttle to the space station), a recursive identification algorithm which can run in real time is preferred.

Closely Spaced Frequencies

In addition to on-orbit testing, the large number of closely spaced modal frequencies makes identification difficult (ref. 12). The algorithm must be able to resolve frequencies that are very close together. It also must be able to handle tens of modes without exceeding the computer facility's capacity.

1.2 Thesis Goal

Identification of unknown systems is a common problem and has been extensively studied. However, LSSs are new and identification of a system with their characteristics has only recently been examined. Two areas which have not been addressed - optimal inputs and processing with limited computational resources - are the subject of this research.

Chapter 2 starts with an overview of what is required to identify a structure (choose a model, an algorithm, and inputs). This procedure is then applied to a LSS. A baseline algorithm is selected -- recursive lattice least squares -- described, and evaluated.

The baseline algorithm was chosen partly on the basis of cycling faster than other algorithms for LSSs. In Chapter 3, a suboptimal version of the baseline algorithm is described which reduces the cycle time even more. This suboptimal version is applied to some simulated LSSs and compared to the baseline version.

One of the conclusions reached in Chapter 3 is that choice of input is very important. This is explored in detail in Chapter 4. A cost function is chosen which is appropriate to a LSS identification experiment, and then minimized to determine an optimal input. The characteristics of the resulting inputs are examined. The effectiveness of the system identification using optimal inputs is compared to using other inputs.

The examples used in chapters 2,3, and 4 to simulate large space structures are slinkys, i.e. masses connected by linear springs and proportional viscous dampers as shown in Figure 1.2.1.

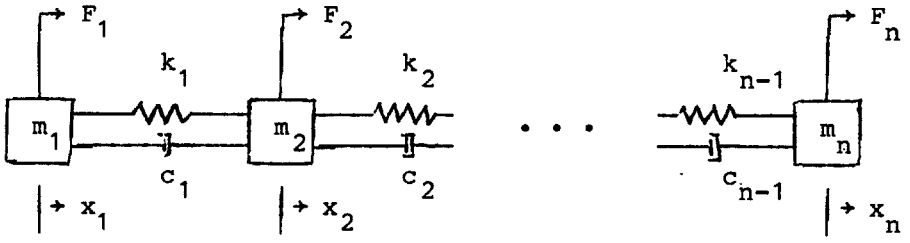


Figure 1.2.1: N-mass Slinky

In all cases the masses are free to move at both ends so that there is always a rigid body mode. In Chapter 5, a more complicated model is used -- a finite element model of the U.S. dual keel space station. The algorithms of Chapters 2 and 3 and the inputs of Chapter 4 are applied to the space station model.

Finally, Chapter 6 summarizes the conclusions reached in the preceding chapters and suggests directions for future work.

Chapter 2: Identification Algorithms

Identification means defining a model, from a specified class of models, which is most nearly equivalent to the system of interest.[13] The choice is based on the results of applying an identification algorithm to inputs and outputs of the system. Thus, identification requires choosing a class of models, an algorithm, and appropriate inputs.

There are almost as many model/algorithm/input combinations as there are identification problems[22]. Thoroughly examining all the possibilities is impossible to do in a few pages, so the approach taken here is to briefly discuss each major decision in the process of picking a model/algorithm/input package in the context of LSSs, then continuing to develop only those possibilities with the most merit.

2.1 Choosing a Model

1. Use a linear model

The first choice is between a linear or nonlinear model. A LSS is a nonlinear, distributed parameter system. However, linear models have worked well in predicting the behavior of flexible structures[9]. Also, identification of linear systems is usually easier to implement than a nonlinear approach. There is no general solution to a nonlinear identification problem - an algorithm effective on one nonlinear system often cannot be implemented on other nonlinear systems. Finally, the identified system is usually used in a controller to compensate for disturbances. A linear model is appropriate for a nonlinear system when excursions are small.

2. Use a lumped parameter (parametric) model

Again, even though a LSS is a distributed parameter (non-parametric) system, lumped parameter models have worked well in describing flexible structures[9]. LSS identification has two goals - verifying the ground-based analysis of the LSS and providing a model the control system can use. Ground-based analysis usually includes a finite element model of the structure, which can be easily

compared to a lumped parameter model. Modern control algorithms use a state-space form, which is a lumped parameter form. (Typical distributed parameter models are impulse responses, sets of partial differential equations, and spectral densities.)

3. Use a time-invariant model

This choice is marginal but turns out not to be critical when combined with all the other choices. The identification algorithm as recommended in this thesis applies as well to a slowly time-varying system as to a time-invariant system.

LSS characteristics vary slowly with time. Temperature changes cause the structure to expand, contract, and warp. Outgassing can change the material properties of a structure. Variations in atmosphere due to fluctuations in the earth environment can affect the amount of damping a structure experiences. To avoid excessive stress on the structure, LSSs are being designed to minimize the effect of temperature variations. The other disturbances should cause only small changes in structural properties and/or take place over a time scale much longer than required for an accurate identification. So a time-invariant model should work well except across major reconfigurations of a space structure, either intentionally or due to a failure. Reconfiguration will be either known ahead of time or detected by failure monitors and can be handled by reinitializing the algorithm.

4. Use a discrete time model

Modern control systems operate in discrete time, as do typical input/output data collectors.

Parameterizations: SISO

The class of models has been restricted to linear, time-invariant, lumped parameter, and discrete time. This is the class of discrete time state space models and all equivalent parameterizations. Parameterization refers to which parameters describing a class of models are assumed to be known. Equivalent parameterizations are ones which can't be distinguished on the basis of input/output data alone.

Some examples of this in the class of discrete time single-input, single-output (SISO) state space models (ignoring noise to simplify the examples) are:

SISO Parameterization 1: All parameters unknown

Example: (9 unknowns)

$$\underline{x}(k+1) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \underline{x}(k) + \begin{bmatrix} e \\ f \end{bmatrix} u(k)$$

$$y(k) = [g\ h]\ \underline{x}(k) + iu(k)$$

SISO Parameterization 2: Observable Canonical Form

Example: (5 unknowns)

$$\underline{x}(k+1) = \begin{bmatrix} 0 & 1 \\ a_1 & b_1 \end{bmatrix} \underline{x}(k) + \begin{bmatrix} c_1 \\ d_1 \end{bmatrix} u(k)$$

$$y(k) = [1\ 0]\ \underline{x}(k) + iu(k)$$

where $a_1 = bc-ad$; $b_1 = a+d$; $c_1 = eg+fh$; $d_1 = (ae+bf)g + (ce+df)h$

SISO Parameterization 3: ARMAX (Auto Regressive Moving Average Exogeneous input) Form

Example: (5 unknowns)

$$y(k) = b_1 y(k-1) + a_1 y(k-2) + iu(k) + d_2 u(k-1) + e_2 u(k-2)$$

where $d_2 = c_1 - ib_1$; $e_2 = d_1 - b_1 c_1 - ia_1$

The systems considered in this thesis are restricted to observable, controllable systems. (Since that covers all the likely structures, it is not a severe restriction.) For such systems, all three of these parameterizations exist and are equivalent. Parameterization 1 is typical of a model derived from a physical knowledge of the system. Parameterization 3 corresponds to the format an identification algorithm uses because the intermediate state $\underline{x}$ has been eliminated. Parameterization 2 is a combination. It has the state

space form of parameterization 1 but the same number of unknowns as parameterization 3.

There are many more parameterizations than the three mentioned here (section 4-3 and Chapter 7 of ref. 6). These are a representative sample and are used in the rest of this thesis.

The number of unknowns is a critical factor. All the unknowns in parameterization 3 can be uniquely determined from input/output data. A unique value of the parameters in parameterization 2 can also be determined from the input/output data, because the number of unknowns is the same as in parameterization 3 and the two sets of unknowns are related by an invertible transformation. However, values for the unknowns in parameterization 1 cannot be uniquely determined solely on the basis of input/output data. The transformation from parameterization 1 to parameterization 2 is not invertible.

Parameterizations: MIMO

The same comments apply to a multi-input, multi-output (MIMO) system. However, specifying the parameterization is more complicated because the coefficients in parameterization 3 become matrices.

For SISO systems, the minimum number of unknowns is always 2N+1 (ref. 14, p. 14), where N=dim($\underline{x}$). N coefficients are associated with N past values of y, and N+1 with the current and N past values of u. Once N is determined, the parameterization can be completely specified.

For a MIMO system, parameterizations 2 and 3 become:

MIMO Parameterization 2: Observable Canonical Form[13]

$$\underline{x}(k+1) = [A_{ij}]\,\underline{x}(k) + B\,\underline{u}(k)$$

$$\underline{y}(k) = \begin{bmatrix} 1 & 0 & \cdots & & & 0 \\ 0 & \cdots & 0 & 1 & 0 \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & \cdots & & 0\ 1\ 0 & \cdots\ 0 & \end{bmatrix} \underline{x}(k) + D\underline{u}(k)$$

column 1 — column ν_1+1

where: $A_{ii} = \begin{bmatrix} 0 & & \\ \vdots & I_{\nu_i - 1} & \\ 0 & & \\ a_{ii1} & \cdots & a_{ii\nu_i} \end{bmatrix}$

$$A_{ij} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ \vdots & \cdots & \vdots \\ \vdots & & \vdots \\ a_{ij1} & \cdots a_{ij\nu_{ij}} & 0 \cdots 0 \end{bmatrix}$$

B = fully populated

D = fully populated

$\nu_{ii} = \nu_i$

$$\nu_{ij} = \begin{cases} \min(\nu_j, \nu_i + 1) & j < i \\ \min(\nu_j, \nu_i) & j \geq i \end{cases} \qquad i,j = 1 \text{ to } p$$

p = number of outputs

The structural indices ν_i are characteristic of the physical system, i.e. are invariant under linear transformation. ν_i equals the number of rows in the observability matrix corresponding to sensor_i when the rows not required for a full rank matrix are deleted[13]. Since the dimension of A_{ii} is $\nu_i \times \nu_i$ (or equivalently since the rank of the observability matrix is N):

$$\sum_{i=1}^{p} \nu_i = N$$

Thus, the A matrix in the observable canonical form of a SISO system is the special case of the A matrix in the observable canonical form of a MIMO system with p=1.

MIMO Parameterization 3: ARMAX form[13]

$$y_i(k) = \underline{a}_{i\nu_i+1}^T \underline{y}'(k) + \underline{a}_{i\nu_i}^T \underline{y}(k-1) + \cdots + \underline{a}_{i1}^T \underline{y}(k-\nu_i) + \underline{d}_i^T \underline{u}(k)$$

$$+ \underline{\beta}_{i\nu_i}^T \underline{u}(k-1) + \cdots + \underline{\beta}_{i1}^T \underline{u}(k-\nu_i)$$

where: $\underline{y}'(k) = [y_1(k) \;\; y_2(k) \cdots y_{i-1}(k) \; y_{i+1}(k) \cdots y_p(k)]^T$

$\underline{a}_{ij}^T = [a_{i1j} \cdots a_{ipj}]$ $\quad i=1$ to p; $j=1$ to ν_i+1

$\underline{\beta}_{ij}^T = -\underline{a}_{ij}^T D + [\gamma_{\nu_1+\cdots+\nu_{i-1}+j,1} \cdots \gamma_{\nu_1+\cdots+\nu_{i-1}+j,\,m}]$

$[\gamma_{ij}] = MB$

$$M = \left[\begin{array}{ccc|c|ccc|c}
-a_{112} & \cdots & -a_{11\nu_1} & 1 & & -a_{1p2} & \cdots & -a_{1p\nu_{1p}} \\
-a_{113} & & & & & & & \\
\vdots & & & & & & & \\
-a_{11\nu_1} & & & & & -a_{1p\nu_{1p}} & & \\
1 & & & 0 & & 0 & & 0 \\
\hline
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
\hline
-a_{p12} & \cdots & -a_{p1\nu_{p1}} & 0 & & -a_{pp2} & \cdots & -a_{pp\nu_p} \\
\vdots & & & & & \vdots & & \\
-a_{p1\nu_{p1}} & & & & & -a_{pp\nu_p} & & \\
0 & & & 0 & & 1 & & 0
\end{array}\right]$$

m = number of inputs

At first glance, there appear to be N×(p+m) + p(m+p-1) unknown coefficients. Each y_i equation has ν_i+1 $\underline{a}$ vectors, the first of which has p-1 elements (because $\underline{y}'$ has p-1 elements) and the rest of which have p elements, one m element $\underline{d}$ vector, and ν_i $\underline{\beta}$ vectors, each of which has m elements:

$$\text{number} = \sum_{i=1}^{p} (\nu_i \times p + p-1 + m + \nu_i \times m) = \left(\sum_{i=1}^{p} \nu_i\right)(p+m) + p(p+m-1)$$

This count overestimates the number of coefficients because some of them must be zero. From the definition of ν_{ij}, $a_{ij\nu_i+1}=0$ for all $j < i$. If all the ν_i are equal, then $a_{ij\nu_i+1}=0$ for all i,j (equivalently, $\underline{a}_{i\nu_i+1} = \underline{0}$ for all i). In that case:

$$\text{MIMO number of coefficients when all the } \nu_i \text{ are equal} = N(p+m) + p \times m \qquad (2.1.1)$$

The structure of the A_{ij} shows that this is the largest number of coefficients in the ARMAX form for any system. It corresponds to the bottom row of each A_{ij} block being full of non-zero elements. For some systems, the number of coefficients may be less. If the sensors have been placed so that some provide very little new information while others provide a lot, the ν_i will be very different and the number of coefficients will be less than that given above. Note that this makes the problem more difficult, not less. The computation time per update is dominated by the y_i (i=1 to p) equation with the largest number of terms i.e. the largest ν_i. The minimum maximum ν_i occurs for all the ν_i equal (or different by no more than one when N is not evenly divisible by p).

Modal Forms

For LSSs, a modal form is a particularly useful parameterization. It is easier to connect the unknowns in a modal form with physically significant quantities than the unknowns in either of the MIMO parameterizations. This connection is made by creating the modal forms from the continuous time physical model (usually a finite element model), then converting to discrete time to get the discrete time equivalent. The forms exist for discrete time systems, but the parameters lose their physical interpretation.

Physical System Model:

$$\dot{\underline{x}}(t) = F\,\underline{x}(t) + G\,\underline{u}(t)$$

$$\underline{y}(t) = H\,\underline{x}(t) + D\,\underline{u}(t)$$

where:

$$F = [\,F_{ij}\,],\quad i=1 \text{ to } n,\ j=1 \text{ to } n$$

$$F_{ii} = \begin{bmatrix} 0 & 1 \\ -[M^{-1}K]_{ii} & -[M^{-1}C]_{ii} \end{bmatrix}$$

$$F_{ij} = \begin{bmatrix} 0 & 0 \\ -[M^{-1}K]_{ij} & -[M^{-1}C]_{ij} \end{bmatrix}$$

M = mass matrix from the finite element model $M\,\ddot{\underline{x}} + C\,\dot{\underline{x}} + K\,\underline{x} =$ force, where $\underline{x}$ is a vector of the nodal displacements of dimension n. The $\underline{x}$ in the Physical System Model contains the nodal displacements and velocities and has dimension N.

C = damping matrix from the finite element model

K = stiffness matrix from the finite element model

G = matrix with odd rows all 0 and even rows determined by actuator placement. The elements of the rows typically equal a known calibration factor divided by a modal mass (there are n modal masses).

H = contains zeroes and known calibration factors which are determined by sensor type and placement

D = contains zeroes and known calibration factors which are determined by sensor type and placement

Two modal forms are in common use, and a third one was derived by modifying the technique used to get MIMO Parameterization 2:

Modal Parameterization 1:

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t)$$

$$\underline{y}(t) = C\,\underline{x}(t) + D\,\underline{u}(t)$$

where:

$$A = \text{diag}\begin{bmatrix} \beta_k & 0 \\ 0 & \beta_k^* \end{bmatrix},\quad k=1 \text{ to } n$$

β_k = eigenvalues of $F = \omega_k * (-\zeta_k + i\sqrt{1-\zeta_k^2})$, k=1 to n. If the kth mode is real, then β_k is a real number, possibly different from the real number β_k.

ζ_k = damping ratio of kth mode

ω_k = undamped vibration frequency of kth mode

B = fully populated = $\Phi_R^{-1} G$

Φ_R = matrix of right eigenvectors of F. The inverse matrix equals (within normalization factors) Φ_L^T, the matrix of left eigenvectors i.e. eigenvectors of F^T

C = fully populated = H Φ_R

D = as in the Physical System model

Φ_R = matrix of (right) eigenvectors of F

The number of non-trivial parameters needing to be identified is N from A, N*m from B, N*p from C, and m*p from D, for a total of N*(m+p+1) + m*p. Note that this is N greater than the total for MIMO Parameterization 3 as given in equation 2.1.1, so that this is not a minimal parameter form. It could be made into a minimal parameter form by choosing the N arbitrary normalization factors on the columns of Φ_R so as to make N elements of C trivial. The physical significance of the unknown parameters is that N are associated with modal damping ratios and frequency and N*(m+p-1) + m*p with elements of the eigenvectors combined with sensor and actuator calibration factors and modal masses. If the sensor calibration factors are such that the sensors are position sensors which sense exactly one state in the Physical System Model system, then the elements of each column of C correspond to (within a normalization factor) elements, at the sensor positions, of the eigenvector of the mode associated with that column. (The two columns of C associated with a vibratory mode are complex conjugates of each other.) Note that this is true even if the modal form was not constructed directly from the Physical System Model. If the system is equivalent to (i.e. a linear transformation of) the Physical System Model, the same procedure can be used to recover the eigenvector elements of the Physical System Model. (The position elements of an eigenvector are collectively referred to as the mode shape.)

If the sensors had been velocity sensors, the column elements would have been corresponding eigenvector elements. Note that while adding velocity sensors increases the number of unknowns to be identified, no new information is obtained. The form of the Physical System Model requires the velocity elements of eigenvectors to be β_k times the position elements, so that determining the position elements (mode shape) of an eigenvector determines the entire eigenvector. The maximum number of useful (from the point of view of providing non-redundant information) sensors is N/2, with each sensor being either a position or velocity sensor and no co-located sensors.

Similarly, the rows of B are elements of the left eigenvectors divided by the modal masses.

A disadvantage of this form is that A, B, and C are complex.

Modal Parameterization 2:

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t)$$

$$\underline{y}(t) = C\,\underline{x}(t) + D\,\underline{u}(t)$$

where:

$$A = \begin{cases} \text{diag}\begin{bmatrix} \text{Re}(\beta_k) & \text{Im}(\beta_k) \\ -\text{Im}(\beta_k) & \text{Re}(\beta_k) \end{bmatrix} & k=1 \text{ to n except for rigid body modes} \\ \text{diag}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} & \text{rigid body modes} \end{cases}$$

B = fully populated = Φ^{-1} G

$\Phi = [\ \text{Re}(\underline{\phi}_1)\ \ \text{Im}(\underline{\phi}_1)\ \ \cdots\ \ \text{Re}(\underline{\phi}_n)\ \ \text{Im}(\underline{\phi}_n)]$

$\underline{\phi}$ = eigenvector of F. For real modes, $\text{Im}(\underline{\phi}_n)$ becomes the eigenvector for the second real root. For rigid body modes, $\text{Im}(\underline{\phi}_n)$ is the generalized eigenvector for the mode.

C = fully populated = H Φ

D = as in the Physical System Model

The number of non-trivial parameters needing to be identified is the same as for Modal Parameterization 1. The same comments on specifying the eigenvector normalizations in Modal Parameterization 1 apply to this form. The procedure requires treating the two columns associated with a vibratory mode as a single complex column and normalizing them simultaneously.

The recovery of eigenvector elements for vibratory modes is a little different from Modal Parameterization 1. The first column of C associated with that mode becomes the real part of the eigenvector element, and the second column becomes the imaginary part.

A disadvantage of this form is that in the limit as $\omega_k \to 0$ (i.e. $\beta_k \to 0$, a rigid body mode), the A matrix abruptly changes form. This makes comparison of regulator performance between a truth model with rigid body modes and an identified model with frequencies at (say) .01 rad/sec difficult to interpret because the state weights are different.

Modal Parameterization 3:

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t)$$

$$\underline{y}(t) = C\,\underline{x}(t) + D\,\underline{u}(t)$$

where:

$$A = \begin{cases} \text{diag}\begin{bmatrix} 0 & 1 \\ -\omega_k^2 & -2*\zeta_k*\omega_k \end{bmatrix} & k=1 \text{ to } n \text{ except for real modes} \\ \text{diag}\begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} & \text{real modes (eigenvalues} = .5*b + \sqrt{.25*b^2 + a}) \end{cases}$$

B = odd rows of first column are 0 and one element of every even row is 1; the rest is fully populated = T G

T = transformation described in appendix B

C = fully populated = $H\,T^{-1}$

D = same as in Physical System model

Because (effectively) one column of B is trivial, this form has the same number of unknowns as Modal Parameterizations 1 and 2, with the normalization already included. In addition, if the mode shapes are real ($M^{-1}K$ and $M^{-1}C$ in the finite element model are diagonalizable by the same matrix) then the odd rows of every column of B are 0.

The disadvantage of this form is that it is more complicated to implement than the other two and there is no simple way to recover mode shapes. It has the conceptual advantage of being in the form used in finite element models. It is the only one of the three forms which always has some elements go to zero when the modes are real.

2.2 Choosing an Algorithm

The parameterization used depends on the algorithm chosen. Even with the class of models limited to linear discrete time state space,

there are many algorithms from which to choose. Most of these can be eliminated because they are inappropriate for LSS identification:

1. Use a recursive (on-line) algorithm

As discussed in the introduction, recursive (on-line) algorithms are preferred to batch (off-line) algorithms because data storage requirements are much less and less powerful computers are required for large systems. Most batch algorithms have a recursive version which has asymptotically the same accuracy. Typically, reasonably good estimates are available much more quickly with a recursive algorithm.

2. The algorithm must work when applied to an operational LSS

As discussed in the introduction, LSS identification will be performed on the operational system. For the low frequency modes of a LSS (periods on the order of 6 seconds), identification can take several minutes. Some control action may be required during an identification experiment, especially if the structure has been excited enough to allow modal identification. The technique must allow the superposition or mixing of inputs good for identification and inputs required to keep the structure stabilized. Any technique, such as Ibrahim Time Domain[15] or Eigensystem Realization[16], which requires a system in free decay after an initial excitation, is not a good choice for orbital testing. A technique like Multi-Shaker Sine Dwell[7], which excites the structure with a pure sinusoid, is even more likely to require control action (due to resonance), and would be inappropriate.

3. Use a time domain technique

Frequency domain techniques (such as Sine Dwell) identify modes by increased structural response at the modal frequency, making closely spaced modes hard to separate. Time domain techniques are preferred for LSSs.

The algorithms consistent with the above choices are recursive least squares (RLS), recursive maximum likelihood (RML), recursive extended least squares (RELS), recursive generalized least squares (RGLS) and their variations. All these algorithms share the

characteristic that they don't assume proportional damping i.e. the modes can be coupled through lumped dampers (as is found in structural joints). This was not considered an important enough property to be used to eliminate algorithms (as proportional damping models have worked well in the past), but it is certainly desirable.

All of these techniques apply to a system in the form of parameterization 3. Since some of the techniques differ in the way the noise is handled, write parameterization 3 in the form:

$$\underline{y}(t) = A\underline{y}(t) + B\,\underline{u}(t) + CD^{-1}\,\underline{e}(t)$$

where: A = matrix whose elements are polynomials in inverse powers of the discrete time shift operator q, with leading term $a_{ij1}\,q^{-1}$

B = matrix whose elements are polynomials starting with b_{ij0}

C = matrix whose elements are polynomials starting with $1+c_{ij1}\,q^{-1}$

D = matrix whose elements are polynomials starting with $1+d_{ij1}\,q^{-1}$

$\underline{e}(t)$ = white noise sequence

The framework which will be used to describe the above algorithms is:

$$\hat{y}_i(t) = \hat{\underline{\theta}}_{Ri}(t-1)\;\underline{\phi}_i(t) \tag{2.2.1}$$

$$\underline{k}_i(t) = \frac{P_i(t-1)\;\underline{\psi}_i(t)}{\lambda(t)\;V_i + \underline{\psi}_i^T(t)\;P_i(t-1)\underline{\psi}_i(t)} \tag{2.2.2}$$

$$P_i(t) = \frac{1}{\lambda(t)}\left[I - \underline{k}_i(t)\;\underline{\psi}_i^T(t)\right]P_i(t-1) \tag{2.2.3}$$

$$\hat{\underline{\theta}}_{Ri}(t) = \hat{\underline{\theta}}_{Ri}(t-1) + \left(y_i(t) - \hat{\underline{\theta}}_{Ri}(t-1)\;\underline{\phi}_i(t)\right)\underline{k}_i^T(t) \tag{2.2.4}$$

where θ = matrix of unknowns being estimated, arranged consistent with the elements of $\underline{\phi}_i$

$\hat{\underline{\theta}}_{Ri}$ = ith row of matrix $\hat{\theta}$

$\underline{\phi}_i$ = vector of regressors for output i which depends on the algorithm

$\underline{k}_i$ = gain vector, one for each of the i outputs

P_i = covariance matrix for the coefficients associated with output i

$\lambda(t)$ = forgetting factor ($0<\lambda\leq 1$, usually near 1)

V_i = variance of noise on the ith sensor. If unknown, it can be estimated with $V_i(t)=V_i(t-1)+\gamma(t)(\varepsilon_i^2(t)-V_i(t-1))$ and $\gamma(t) = \frac{\gamma(t-1)}{\gamma(t-1)+\lambda(t)}$ with $\gamma(-1)$ = large (user choice)

$\underline{\psi}_i$ = derivative (or its approximation) of the estimate of output i with respect to its unknowns = $\frac{d^T[\hat{y}_i(t)]}{d\underline{\theta}_i}$. Of the algorithms presented in this section, RELS and RGLS use an approximation for the derivative.

i = 1 to p

To simplify the summary of the algorithms, "coefficients in A" will be used to indicate a matrix whose elements are the a_{ijk} of the MIMO parameterization 3 form. The polynomial elements of the matrix A have been spread out into a larger matrix whose elements are the coefficients of the polynomials. The leading "ones" in the C and D matrix polynomials are not included in their expanded matrices. For more detail, see ref. 20, Chap. 3.

Recursive Least Squares (RLS)

Model: $\underline{y}(t) = A\,\underline{y}(t) + B\,\underline{u}(t) + \underline{e}(t)$ i.e. C=D=I

$$\theta^T = [\text{coefficients in A} \quad \text{coefficients in B}]$$

$$\underline{\phi}_i^T = [\underline{y}^T(t-1) \cdots \underline{y}^T(t-\nu_i)\ \underline{u}^T(t) \cdots \underline{u}^T(t-\nu_i)]$$

$$\underline{\psi}_i = \underline{\phi}_i$$

Advantages: (1) Simplest algorithm

(2) Fastest algorithm

(3) Not very sensitive to initial guess for θ and P

Disadvantages: (1) Estimate is biased if C or D differs from I

(2) Convergence is usually slower than RML

Recursive Maximum Likelihood (RML)

I. Model: $\underline{y}(t) = A\,\underline{y}(t) + B\,\underline{u}(t) + C\,\underline{e}(t)$ i.e. D=I (same as RELS)

$\theta^T = [\text{coefficients in A} \quad \text{coefficients in B} \quad \text{coefficients in C}]$

$\underline{\phi}_i^T(t) = [\underline{y}^T(t-1)\cdots\underline{y}^T(t-\nu_i)\ \underline{u}^T(t)\cdots\underline{u}^T(t-\nu_i)\ \underline{w}^T(t-1)\cdots\underline{w}^T(t-\nu_i)]$

$\underline{\psi}_i^T(t) = [\tilde{\underline{y}}^T(t-1)\cdots\tilde{\underline{y}}^T(t-\nu_i)\ \tilde{\underline{u}}^T(t)\cdots(t-\nu_i)\ \tilde{\underline{w}}(t-1)\cdots\tilde{\underline{w}}^T(t-\nu_i)]$

where: $\underline{w}(t) = \hat{C}^{-1}\left((I-\hat{A})\,\underline{y}(t) - \hat{B}\underline{u}(t)\right)$

$\hat{C}\,\tilde{\underline{y}}(t)=\underline{y}(t)$ i.e. $\tilde{\underline{y}}(t)=\underline{y}(t)-[\text{coefficients in } \hat{C}][\tilde{\underline{y}}^T(t-1)\cdots\tilde{\underline{y}}^T(t-\nu_i)]^T$

$\hat{C}\,\tilde{\underline{u}}(t) = \underline{u}(t)$

$\hat{C}\,\tilde{\underline{w}}(t) = \underline{w}(t)$

Advantages: (1) Converges rapidly near the correct solution

(2) Estimate is unbiased for C differing from I

Disadvantages: (1) Calculating C, $\underline{w}$, $\tilde{\underline{y}}$, $\tilde{\underline{u}}$, and $\tilde{\underline{w}}$ costs cycle time

(2) May not converge if initial guess for A, B, and C is poor

II. Model: $\underline{y}(t) = A\,\underline{y}(t) + B\,\underline{u}(t) + D^{-1}\underline{e}(t)$ i.e. C=I (same as RGLS)

$\theta^T = [\text{coefficients in A} \quad \text{coefficients in B} \quad \text{coefficients in D}]$

$\underline{\phi}_i^T(t) = [\underline{y}^T(t-1)\cdots\underline{y}^T(t-\nu_i)\ \underline{u}^T(t)\cdots\underline{u}^T(t-\nu_i)\ \underline{v}^T(t-1)\cdots\underline{v}^T(t-\nu_i)]$

$\underline{\psi}_i^T(t) = [\tilde{\underline{y}}^T(t-1)\cdots\tilde{\underline{y}}^T(t-\nu_i)\ \tilde{\underline{u}}^T(t)\cdots\tilde{\underline{u}}^T(t-\nu_i)\ \tilde{\underline{v}}^T(t-1)\cdots\tilde{\underline{v}}^T(t-\nu_i)]$

where: $\underline{v}(t) = \hat{D}\left((I-\hat{A})\,\underline{y}(t) - \hat{B}\,\underline{u}(t)\right)$

$\tilde{\underline{y}}(t) = \hat{D}\underline{y}(t)$

$\tilde{\underline{u}}(t) = \hat{D}\underline{u}(t)$

$\tilde{\underline{v}}(t) = \hat{D}\underline{v}(t)$

Advantages and disadvantages same as for RML (I) with C replaced by D.

Recursive Extended Least Squares (RELS)

Model: $\underline{y}(t) = A\,\underline{y}(t) + B\,\underline{u}(t) + C\,\underline{e}(t)$ i.e. D=I (same as RML (I))

θ = same as RML (I)

$\underline{\phi}_i$ = same as RML (I)

$\underline{\psi}_i = \underline{\phi}_i$

Advantages: (1) Faster than RML

(2) Estimate is unbiased for C differing from I

Disadvantages: (1) Calculating C and $\underline{w}$ costs cycle time

(2) May not converge if initial guess for A, B, and C is poor

(3) Less likely to converge than RML (depends on C)

Recursive Generalized Least Squares (RGLS)

Model: $\underline{y}(t) = A\,\underline{y}(t) + B\,\underline{u}(t) + D^{-1}\underline{e}(t)$ i.e. C=I (same as RML (II))

θ = same as RML (II)

$\underline{\phi}_i$ = same as RML (II)

$\underline{\psi}_i = \underline{\phi}_i$

Advantages and disadvantages same as for RELS with C replaced by D.

All these simplify to RLS if the noise is white (C=D=I), but the noise is almost certainly correlated. Typically, the measurement noise for a system in the form of parameterization 1 is close to white. Transforming to parameterization 3 gives $CD^{-1}=(I-A)$ or C=I-A, D=I. The noise in a real system would have to be correlated in a very specific way to be white in parameterization 3 form. RML(I) or RELS is required for accurate estimates when the noise level is high.

For the noise levels considered in this thesis (signal to noise ratio $\geq$ 100), RLS converged and gave accurate estimates. Since RLS is significantly faster than either RML or RELS, RLS was chosen as the best technique for LSSs. (More accurately, an alternative formulation of RLS, as described in section 2.4.)

2.3 Inputs

Choosing the input is discussed in Chapter 4. During the early work on examining identification algorithms, the inputs used were pseudo-random binary sequences (PRBSs). To make the inputs as realistic as possible, a supervisory control loop was added so that:

1. If the rigid body velocity exceeds a deadband or the rigid body position exceeds a deadband and the rigid body velocity has the same sign, all inputs fire in the same direction (that which decreases the rigid body motion).
2. If 1 isn't in effect, and the flexible component of any sensor exceeds a deadband, no inputs fire.

It is assumed that the controller has perfect knowledge of the rigid body and flexible motion.

After adding the supervisory control, PRBSs became more difficult to use. Some PRBSs worked very well, others did not. As discussed in Chapter 4, this is not surprising, but was difficult to work with, and emphasized the importance of choosing good inputs. Much better results were achieved by using an input which consisted of the sum of several sinusoids with a linear decay envelope (see Chapter 4). To simulate modelling error, the sinusoid frequencies were chosen to be not too close to the true frequencies.

2.4 LSS Algorithm (Recursive Lattice Least Squares (RLLS))

The implementation of the estimation equations for least squares techniques has changed as ways of doing the same calculation more efficiently and/or accurately have been discovered. Gauss[11] (circa 1800) was the first to use least squares (for an orbit determination problem). His technique is now referred to as batch least squares. The Kalman filter (circa 1960) was the first recursive version, as well as being designed for easy use on computers.

Equations 2.2.1-4 are written in the Kalman filter form. Equation 2.2.3 (P equation) was very sensitive to round-off errors due to finite computer word length, leading to the development of a square root version (1963) and UDU^T factorization (1974)[3]. UDU^T factorization is currently the standard implementation of equation 2.2.3, eliminating the numerical problem for the P matrix.

The latest improvement, called a lattice (or ladder) filter, attacks 2 problems, both important for LSSs: choice of system order and the number of computations required per new data point. Although RLLS has been used some for LSS identification[23], it is not widely known outside speech and signal processing. For those unfamiliar with RLLS, the next few pages describe the algorithm.

Derivation

The motivation for the development of RLLS was to obtain an algorithm which didn't require an a priori choice of system order (ref. N, p. 350). Consider the RLLS version of equation 2.2.1:

$$\hat{\underline{y}}^{(n)}(t) = \hat{\theta}_n^{(n)} \underline{\phi}^{(n)}(t-1) \qquad (2.4.1)$$

$$\underline{\phi}^{(n)}(t-1) = \left[\underline{y}^T(t-1)\ \underline{u}^T(t-1)\ \cdots\ \underline{y}^T(t-n)\ \underline{u}^T(t-n)\right]^T \qquad (2.4.2)$$

where superscript (n)=nth order model and subscript n means nth order part (see equations on next page for need for this subscript). Note that in equation 2.2.1 the regressor is $\underline{\phi}(t)$ and in 2.4.1 it is $\underline{\phi}(t-1)$. This is because RLLS (as presented in this thesis) does not allow use of $\underline{u}(t)$ as a regressor i.e. it assumes that D (in $\underline{y}=C\underline{x}+D\underline{u}$) is zero. For LSSs, this is no problem unless un-integrated accelerometer data is being used. In that case, $\underline{u}(t)$ has to be delayed one time step so that $\underline{u}(t)$ is labeled $\underline{u}(t-1)$. This would increase the identifier order by one and would require modifying the equations to recover the ARMA coefficients so that half the coefficients are not shifted one order. In the rest of this thesis, the D matrix is assumed to be zero.

In general, the regressors $\underline{y}(t)$ and $\underline{u}(t)$ are correlated with their values at other times:

$$\begin{bmatrix} \underline{y}(t-n-1) \\ \underline{u}(t-n-1) \end{bmatrix} = H^{(n)}(t-1)\ \underline{\phi}^{(n)}(t-1) + \underline{r}_n(t-1) \qquad (2.4.3)$$

where $\underline{r}_n(t-1)$ is the uncorrelated (orthogonal) part of $\left[\underline{y}^T(t-n-1)\ \underline{u}^T(t-n-1)\right]^T$. Then:

$$\hat{\underline{y}}^{(n+1)}(t) = \hat{\theta}^{(n+1)}_{n+1}\,\underline{\phi}^{(n+1)}(t-1)$$

$$= \hat{\theta}^{(n+1)}_{n}\,\underline{\phi}^{(n)}(t-1) + \begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \underline{y}(t-n-1) \\ \underline{u}(t-n-1) \end{bmatrix}$$

$$= \hat{\theta}^{(n+1)}_{n}\,\underline{\phi}^{(n)}(t-1) + \begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix}\left(H^{(n)}(t-1)\underline{\phi}^{(n)}(t-1) + \underline{r}_n(t-1)\right)$$

$$= \left(\hat{\theta}^{(n+1)}_{n} + \begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix}H^{(n)}(t-1)\right)\underline{\phi}^{(n)}(t-1) + \begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix}\underline{r}_n(t-1)$$

Since $\hat{y}^{(n+1)}(t) = \hat{y}^{(n)}(t)$ + (new information from adding n+1 terms):

$$\hat{y}^{(n)}(t) = \left(\underline{\hat{\theta}}^{(n+1)}_{n} + \begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix}H^{(n)}(t-1)\right)\underline{\phi}^{(n)}(t-1) = \underline{\hat{\theta}}^{(n)}_{n}\,\underline{\phi}^{(n)}(t-1)$$

or: $$\underline{\hat{\theta}}^{(n+1)}_{n} = \underline{\hat{\theta}}^{(n)}_{n} - \begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix}H^{(n)}(t-1)$$

Thus, for RLS, increasing system order changes the estimates of all the parameters, not just the additional ones. Increasing the system order requires restarting the identification.

In RLLS, the regressors $\underline{r}(t)$ described above are used in the identification instead of $\underline{y}(t)$ and $\underline{u}(t)$. Because of the orthogonality property, parameters are identified one order at a time instead of one inputs must be replaced by r as well as the outputs.) Because RLLS replaces the inputs with orthogonal regressors, it requires a regressive model of the inputs as well as the outputs. Thus, RLLS identifies more unknowns than RLS, although the additional parameters are not directly required to reconstruct the unknown system.

The RLLS equations can be derived either as an orthogonal projection of $[\underline{y}^T(t)\ \underline{u}^T(t)]^T$ on $\underline{\phi}(t)$ or a transformation of coordinates (see Appendix A). These equations are listed in Figure 2.4.1.

Extension of Derivation

Four modifications make this algorithm easier to use and less complex:

(1) The equations in Figure 2.4.1 assume that the state variables, inputs, and outputs have been normalized so that the measurement noise has variance one for all sensors. Including the variance explicitly permits using variance = 0 to check that the equations are working properly. (When there is no noise and the RLLS noise variance parameter γ is set to zero, the identifier converges to the true system exactly, in a finite number of steps.) It also allows this scalar parameter to be varied to adjust the convergence rate of the identifier for best performance. In the examples in this thesis, the best performance was for the variance to be set to about 100 times the true variance. The state variables have been normalized so that the true noise variance is the same for all outputs.

(2) Defining $P_n = R_n^{-1}$ and $P_n^* = R_n^{*-1}$ eliminates two matrix inversions, saving computation time for systems with m (=number of inputs) + p (=number of outputs) > 3 (see Figure 2.4.3).

If P_n and P_n^* are used:

(3) The parameter F, which is an intermediate parameter used in the standard form of the algorithm (see Figure 2.4.1), can be eliminated, saving computation time.

(4) A UDU^T formulation for P_n and P_n^* can be used, increasing numerical stability. This costs computation time, but the improvement in numerical stability is usually necessary for good algorithm performance.

The details of incorporating these three changes into the equations are given in Appendix A. The resulting equations are listed in Figure 2.4.2. A comparison of the number of multiplications required for the standard and modified algorithms is presented in Figure 2.4.3. The savings in computation time for typical LSS models is substantial.

Initialization: $F_n(0) = 0$

$R_n(-1) = R_n^*(0) = \text{small I}$

$\beta_o(t) = 1$

Each new measurement: $\underline{e}_o(t) = \underline{r}_o(t) = [\underline{y}^T(t)\ \underline{u}^T(t)]^T$

For n=0, •••, M-1:

$$R_n(t-1) = \lambda R_n(t-2) + \underline{r}_n(t-1)\ \underline{r}_n^T(t-1)/\beta_n(t)$$

$$R_n^*(t) = \lambda\ R_n^*(t-1) + \underline{e}_n(t)\ \underline{e}_n^T(t)/\beta_n(t)$$

$$F_n(t) = \lambda\ F_n(t-1) + \underline{r}_n(t-1)\ \underline{e}_n^T(t)/\beta_n(t)$$

$$K_n(t) = F_n^T(t)\ R_n^{-1}(t-1)$$

$$K_n^*(t) = F_n(t)\ R_n^{*-1}(t)$$

For n≠M-1:

$$\underline{r}_{n+1}(t) = \underline{r}_n(t-1) - K_n^*(t)\ \underline{e}_n(t)$$

$$\underline{e}_{n+1}(t) = \underline{e}_n(t) - K_n(t)\ \underline{r}_n(t-1)$$

$$\beta_{n+1}(t) = \beta_n(t) - \underline{r}_n^T(t-1)\ R_n^{-1}(t-1)\ \underline{r}_n(t-1)$$

where, for $\underline{x}(t) = G(t)\ \underline{\phi}(t-1)$:

$$G^{(n+1)}(t) = [G^{(n)}(t)\ \ 0] + K_n(t)[-H^{(n)}(t-1)\ \ I]$$

$$H^{(n+1)}(t) = [0\ \ H^{(n)}(t-1)] + K_n^*(t)[I\ \ -G^{(n)}(t)]$$

Figure 2.4.1: RLLS Equations as Presented in the Literature

Initialization: $K_n(0) = K_n^*(0) = 0$ $\quad n=0 \to M-1$

$P_n(-1) = P_n^*(0) = \text{large } I$

$\gamma_o(t) = \lambda V$, where λ = forgetting factor

V = measurement noise variance

Each new measurement:

$$\underline{e}_o(t) = \underline{r}_o(t) = \left[\underline{y}^T(t)\ \underline{u}^T(t)\right]^T$$

For $n=0 \to M-1$:

$$\underline{k} = \frac{P_n(t-2)\ \underline{r}_n(t-1)}{\underline{r}_n^T(t-1)\ P_n(t-2)\ \underline{r}_n(t-1) + \gamma_n(t)}$$

$$K_n(t) = K_n(t-1) + \left[\underline{e}_n(t) - K_n(t-1)\ \underline{r}_n(t-1)\right]\underline{k}^T \qquad (2.4.4)$$

$$P_n(t-1) = \frac{1}{\lambda}\left\{I - \underline{k}\ \underline{r}_n^T(t-1)\right\}P_n(t-2)$$

$$\underline{k}^* = \frac{P_n^*(t-1)\ \underline{e}_n(t)}{\underline{e}_n^T(t)\ P_n^*(t-1)\ \underline{e}_n(t) + \gamma_n(t)}$$

$$K_n^*(t) = K_n^*(t-1) + \left[\underline{r}_n(t-1) - K_n^*(t-1)\ \underline{e}_n(t)\right]\underline{k}^{*T} \qquad (2.4.5)$$

$$P_n^*(t) = \frac{1}{\lambda}\left[I - \underline{k}^*\ \underline{e}_n^T(t)\right]\ P_n^*(t-1)$$

For $n \neq M-1$:

$$\underline{e}_{n+1}(t) = \underline{e}_n(t) - K_n(t)\ \underline{r}_n(t-1) \qquad (2.4.6)$$

$$\underline{r}_{n+1}(t) = \underline{r}_n(t-1) - K_n^*(t)\ \underline{e}_n(t) \qquad (2.4.7)$$

$$\gamma_{n+1}(t) = \gamma_n(t) - \lambda\ \underline{r}_n^T(t-1)\ P_n(t-1)\ \underline{r}_n(t-1) \qquad (2.4.8)$$

To recover RLS coefficients ($\underline{x}(t) = G(t)\ \underline{\phi}(t-1)$):

$$G^{(n+1)}(t) = \left[G^{(n)}(t)\ \ 0\right] + K_n(t)\left[-H^{(n)}(t-1)\ \ I\right]$$

$$H^{(n+1)}(t) = \left[0\ \ H^{(n)}(t-1)\right] + K_n^*(t)\left[I\ \ -G^{(n)}(t)\right] \qquad (2.4.9)$$

with: $G^{(1)}(t) = K_o(t)$ and $H^{(1)}(t) = K_o^*(t)$

Figure 2.4.2: Alternate Form of RLLS Equations

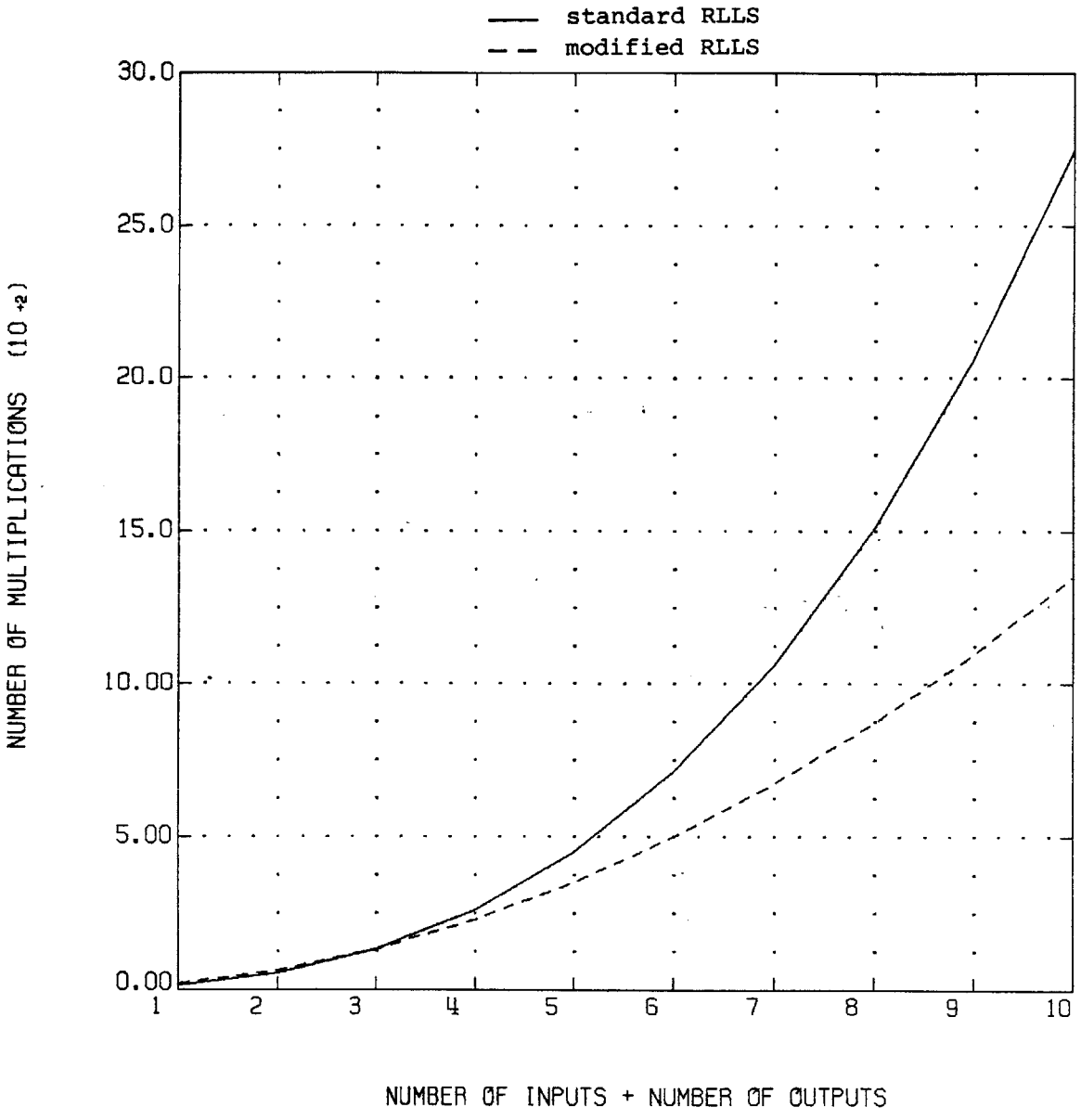


Figure 2.4.3: Comparison of Number of Multiplications Required for Standard and Modified RLLS

Correlated Noise

RLLS has the same problem with correlated noise as RLS i.e. the estimates are biased for correlated noise. Extended least squares can be used in a lattice filter (called RLELS) to eliminate the bias just as it is used in RLS, with $\underline{Ay} + \underline{Bu}$ replaced by $\underline{Kr}$. Equations 2.4.4-9 are unchanged in form, but their dimension is increased because the error estimates $\underline{\varepsilon}_n(t)$ are added to the regressors. The execution order is rearranged to allow calculation of $\underline{\varepsilon}_n(t)$:

Each new measurement:

$\underline{\varepsilon}_o(t) = \underline{y}(t)/\gamma_o(t)$

For n = 0 → M-1:

$\underline{\varepsilon}_{n+1}(t) = \underline{\varepsilon}_n(t) - K_n(t-1)\underline{r}_n(t-1)/\gamma_n(t)$

Do $P_n(t-1)$ equation from equations (2.4.4)

Do equation (2.4.8) (the $\gamma_{n+1}(t)$ equation)

$\underline{e}_o(t) = \underline{r}_o(t) = \left[\underline{y}^T(t) \quad \underline{u}^T(t) \quad \underline{\varepsilon}_n^T(t)\gamma_n(t)\right]^T$

Do equations (2.4.4)-(2.4.7) and (2.4.9), with $P_n(t-1)$ equation from equations (2.4.4) deleted.

Evaluating Performance

Because of the large number of parameters involved, measuring the accuracy of an identification is difficult. In addition, the two basic purposes of an identification have slightly different goals. The modellers need to know where the largest errors in the model are, an indication of where the modelling assumptions are most in error. The users need to know how well a controller based on the identifed model will perform with the real system.

Comparing the numerical values of the identified parameters in the RLLS model is not useful for either group. It is impossible to tell how significant errors in a single parameter are, or whether errors in two parameters tend to cancel each other out. The approach used here is to take the identified system, convert it to modal form, and use the result in a simple control system. The modal form is well suited to revealing identifier errors in the model, and comparing the performance using a simple control system reveals how important the identifier errors are in degrading system performance. The procedure is:

(1) Transform the RLLS model to a RLS (ARMAX) model.

(2) Transform the RLS model to continuous time.

(3) Transform to Modal Parameterization 2 form. The modes are now decoupled.

(4) Perform an eigen analysis on the system matrix to recover the identified eigenvalues (frequencies and damping ratios). Compare with the known correct values.

(5) Recover the identified mode shapes from the C matrix. Compare with the known correct values.

(6) If any rigid body modes were identified as low frequency flexible modes, transform them to Modal Parameterization 3 form. (Since the modes are decoupled this can be done without transforming the other modes.) This step is required to insure that as the frequency of such modes approach 0, the form of the system matrix will approach a rigid body mode.

(7) Transform to discrete time using the step size for the control system.

(8) Use the identified system to generate a steady state linear quadratic regulator with steady state Kalman filter observer. Run the known true system with this identified regulator+filter and look at the LSS transient response, closed loop poles, and standard deviation of the steady state errors. Compare with results for the true system. Note that the identified and true cases use the same weighting and covariance matrices to design the regulator and filter and the same A, B, and C matrices to calculate the true state (those for the true system). They use different gains and different A, B, and C matrices to propagate the state.

Presenting the frequencies, damping ratios, mode shapes, and regulator performance is unnecessary for every example. In general, the more accurate the identification, the closer all these measures are to correct. For brevity, comparison between data runs will be done by comparing identified frequencies. Examination of a single run's accuracy will include all the measures.

Figures 2.4.4-12 examine the performance of RLLS on an 8 mass slinky. The SNR used to generate this data, averaged over the sensors, was about 100. Figures 2.4.4 and 2.4.5 show the time history of the two inputs (on masses 1 and 5). Figure 2.4.6 shows the identified frequencies. The error is 2% or less for all modes. Figure 2.4.7 shows the identified damping ratios. The percentage error is large (318%) because the damping ratios are so small, but the absolute magnitude of the error is small (worst error was .024). Figure 2.4.8 compares the real part of the identified and true mode shapes at the four masses where there were sensors (1,3,5, and 7). Figure 2.4.9 shows the same comparison for the imaginary part (which was zero for the true modes). Figure 2.4.10 compares the regulator commanded inputs for the true and identified systems. The true inputs use gains determined using the true system, and the states are estimated using the true system. The identified system uses the identified system to calculate gains and propagate states, but the exact state is obtained by applying the commanded inputs to the true system. Figure 2.4.11 compares the outputs for the true and identified system+regulator+ filter. Figure 2.4.12 compares the closed loop poles and standard deviation of the steady state errors for the true and identified systems. All these figures indicate that RLLS works very well.

Figure 2.4.13 compares the performance of RLLS, RLELS with forgetting factor=1, and RLELS with forgetting factor=.98. For all modes, RLELS was as good or better than RLLS, but the improvement was slight. The difference in forgetting factor was not significant.

As demonstrated in the previous pages, RLLS is a good baseline algorithm for identifying a LSS. Since it is recursive, the data storage requirements are not large. It requires no specific type of inputs, so a controller running in parallel with the identification experiment to stabilize the structure does not interfere. The resulting identified model is useful both for verifying a finite element model and for use in a linear, discrete time control system. A version exists (RLELS) which provides unbiased estimates for correlated noise, if the extra accuracy is desired over faster cycle times. The errors in frequency and damping ratio in the identifier are acceptable for reasonable run times (Figures 2.4.6 and 7).

Input 1 = .037*cos(.35τ)+.064*cos(.68τ)-.103*cos(.95τ)+.077*cos(1.24τ)+
.096*cos(1.5τ)+.150*cos(1.75τ)+.193*cos(1.9τ)

where τ = time+20 so that the terms don't all start at a peak

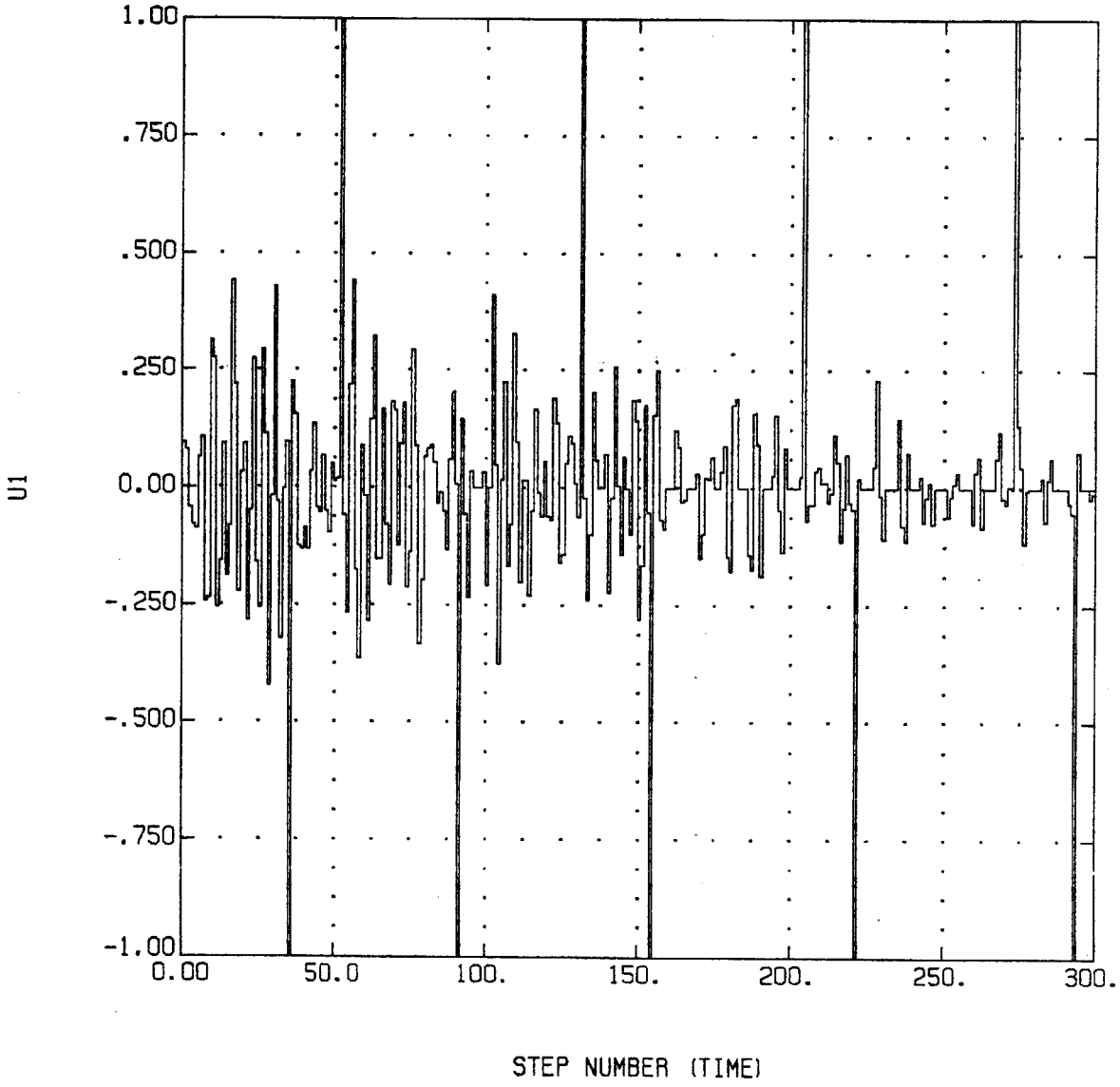


Figure 2.4.4: Input from Actuator 1 During Identification Run

Input 2 =-.125*cos(.43τ)+.074*cos(.84τ)+.151*cos(.99τ)+.113*cos(1.28τ)+
.112*cos(1.54τ)+.144*cos(1.79τ)+.122*cos(1.94τ)

where τ = time+20 so that the terms don't all start at a peak

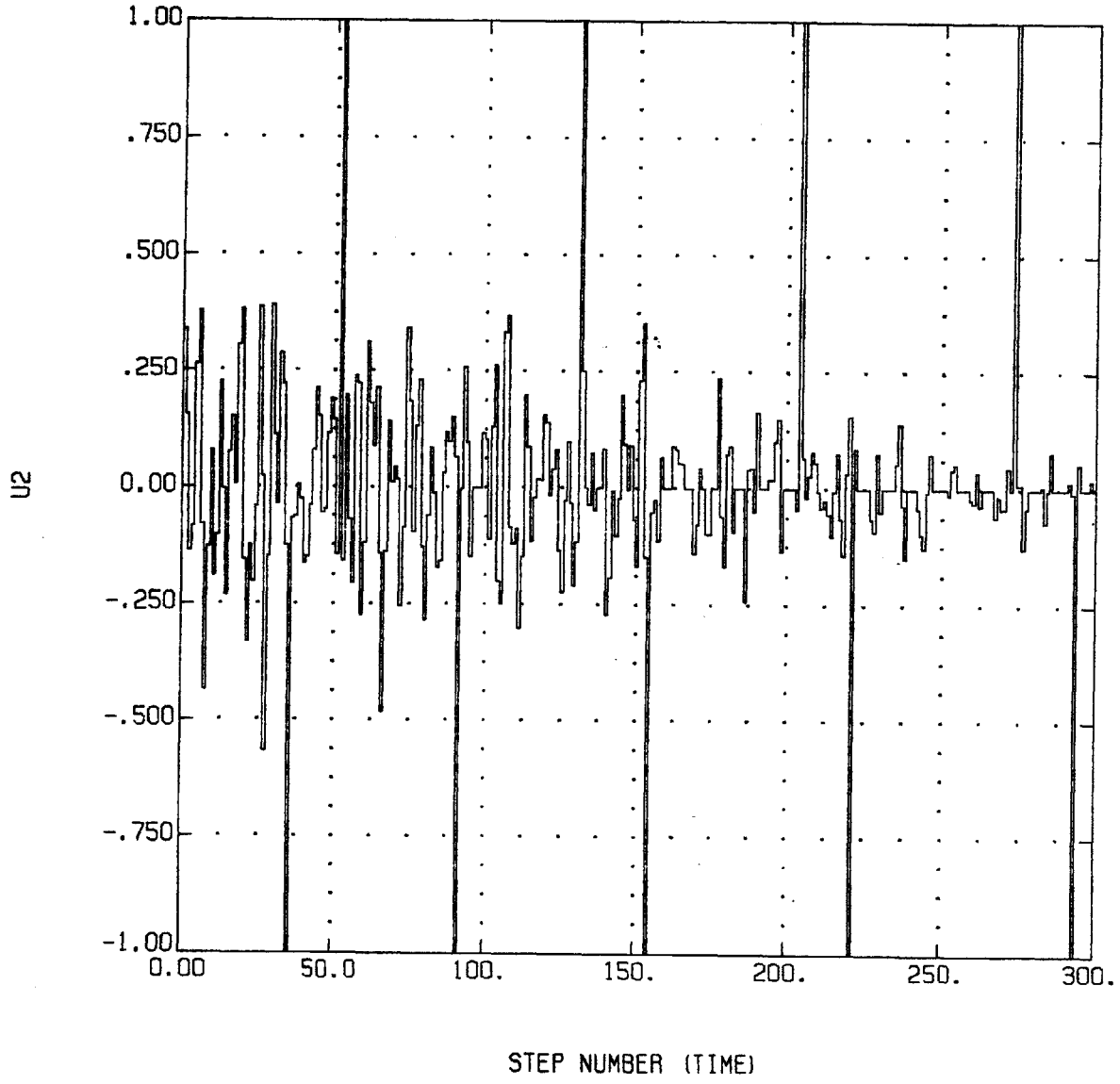


Figure 2.4.5: Input from Actuator 2 During Identification Run

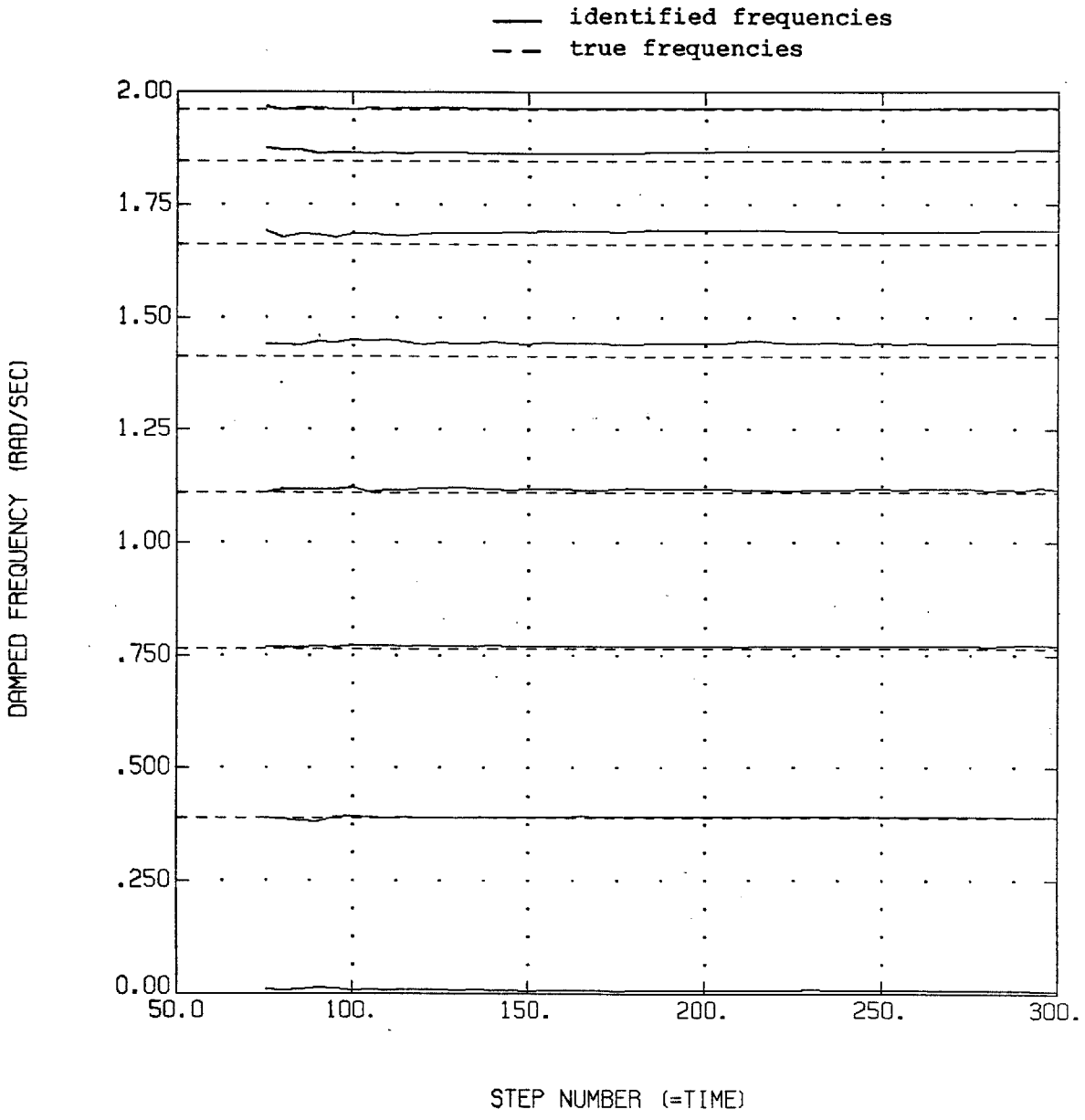


Figure 2.4.6: Identified Frequencies for 8 Mass Slinky

changing lines are identified damping ratios
straight " " true " "

flex mode 1 from Figure 2.4.5
" " 2 " " "
" " 3 " " "
" " 4 " " "
" " 5 " " "
" " 6 " " "
" " 7 " " "

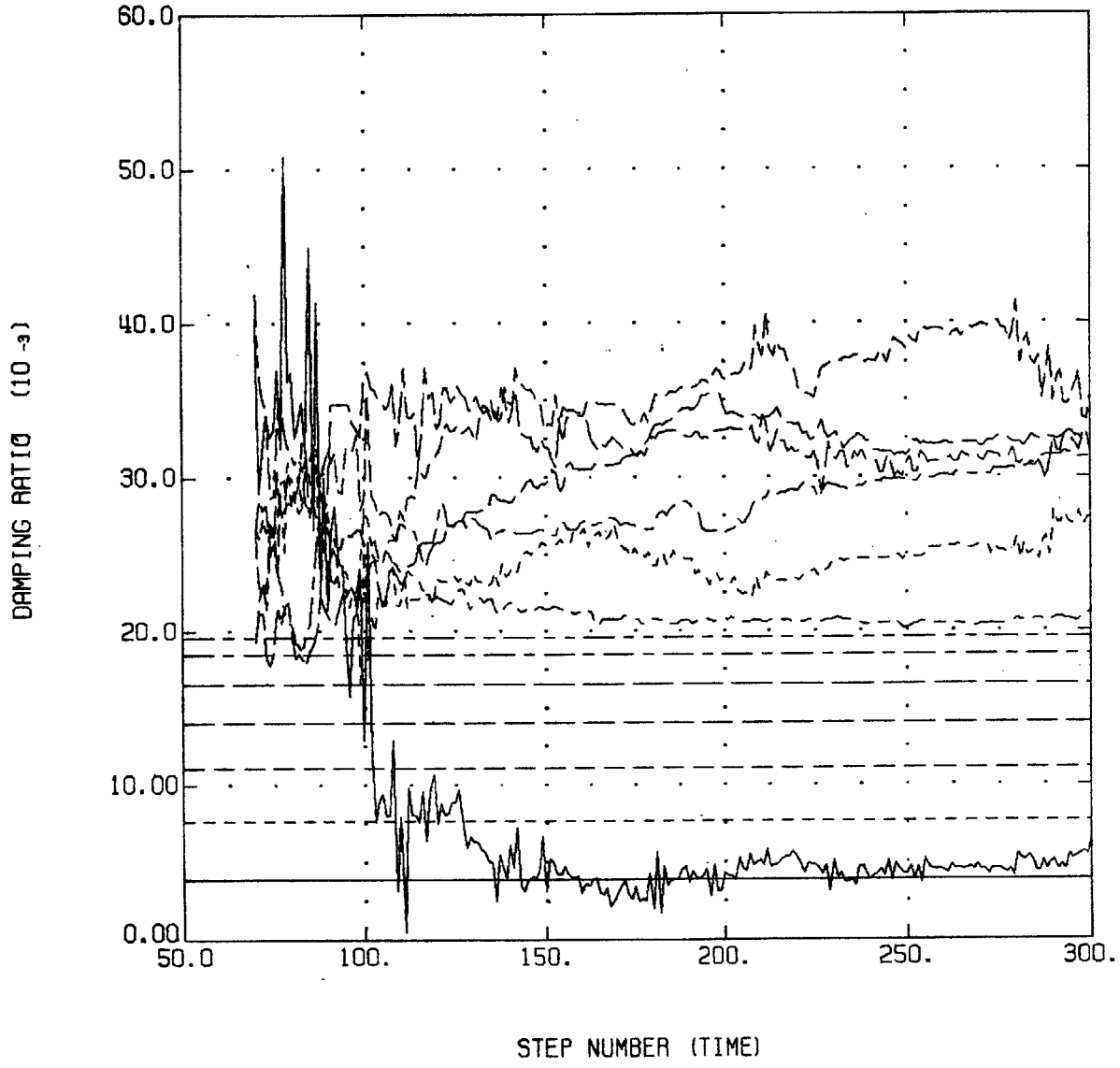


Figure 2.4.7: Identified Damping Ratios for 8 Mass Slinky

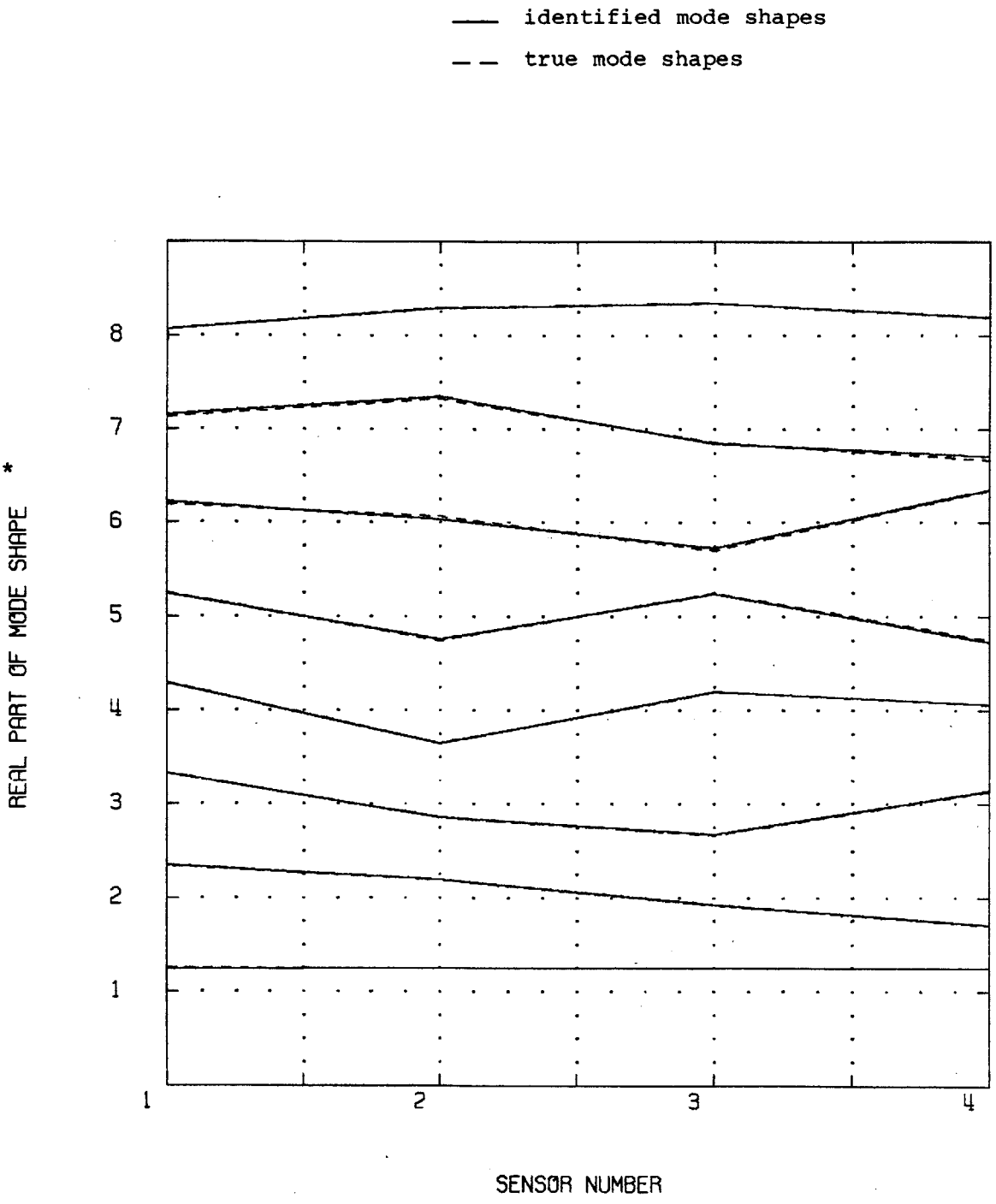


Figure 2.4.8: Identified Mode Shapes for 8 Mass Slinky

*Integers are mode number; deflections about 0 for each integer are the mode shape

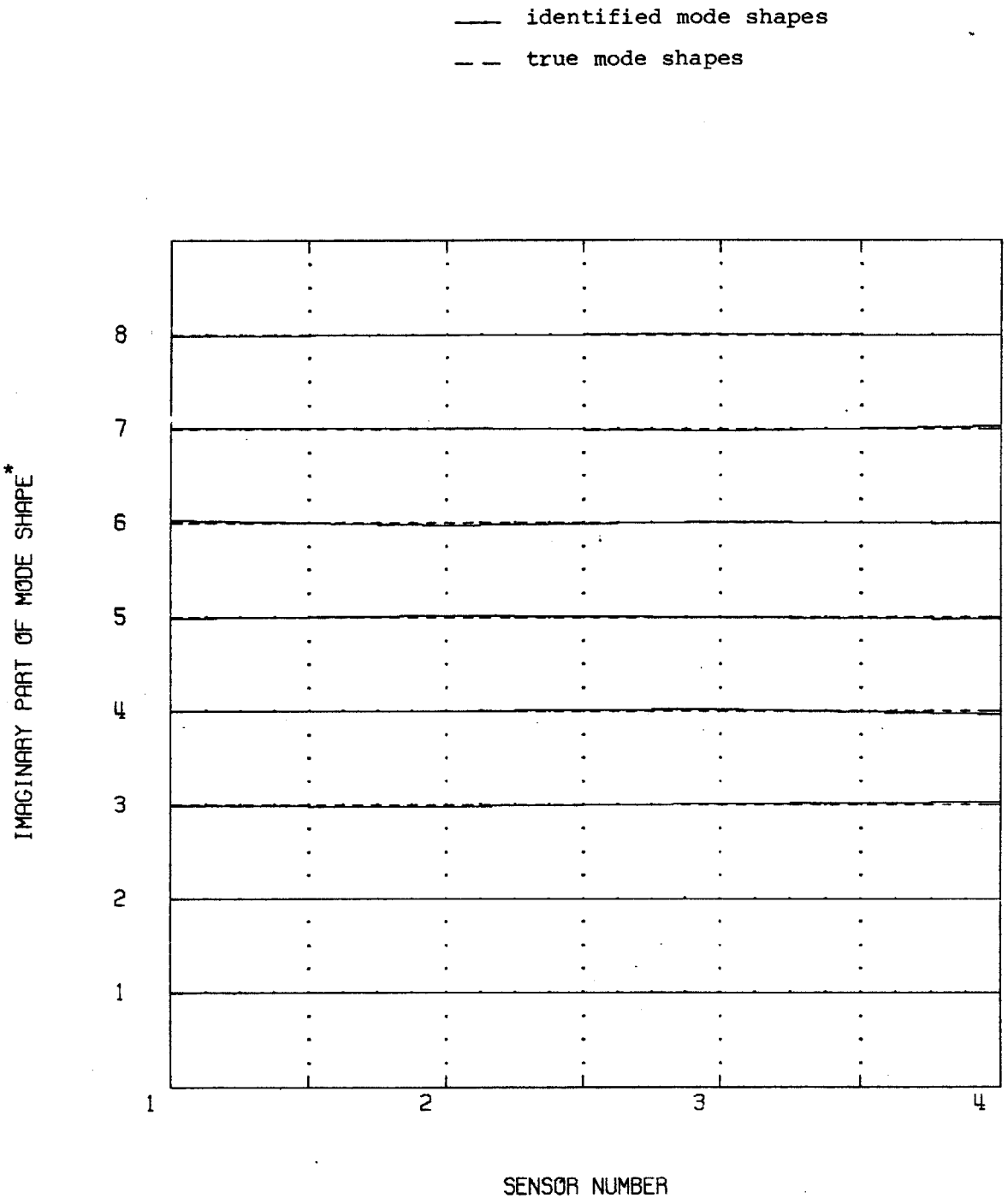


Figure 2.4.9: Imaginary Part of Identified Mode Shapes for 8 Mass Slinky

*Integers are mode number; deflections about 0 for each integer are the mode shape

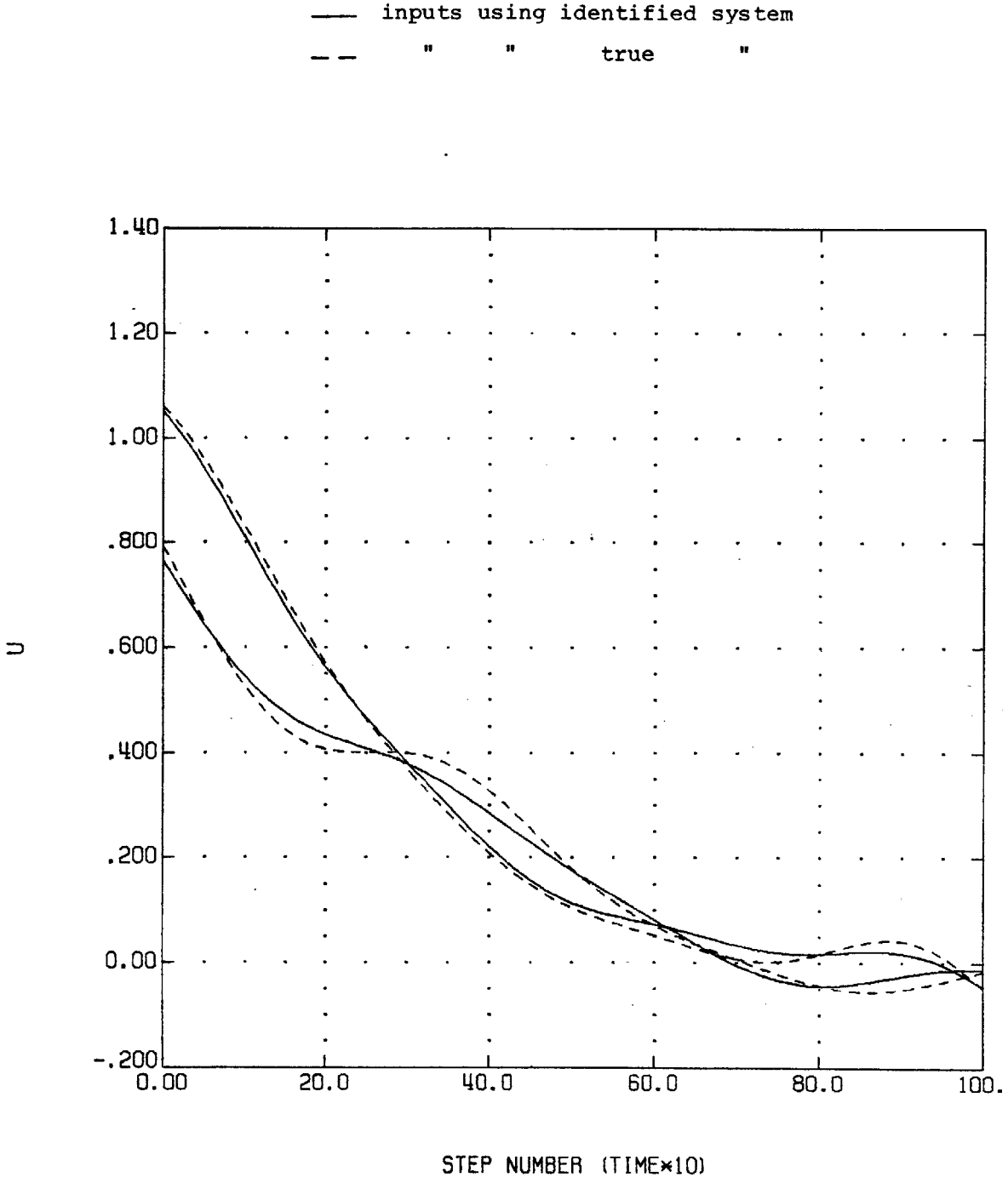


Figure 2.4.10: Control History for Identified Closed Loop System for 8 Mass Slinky

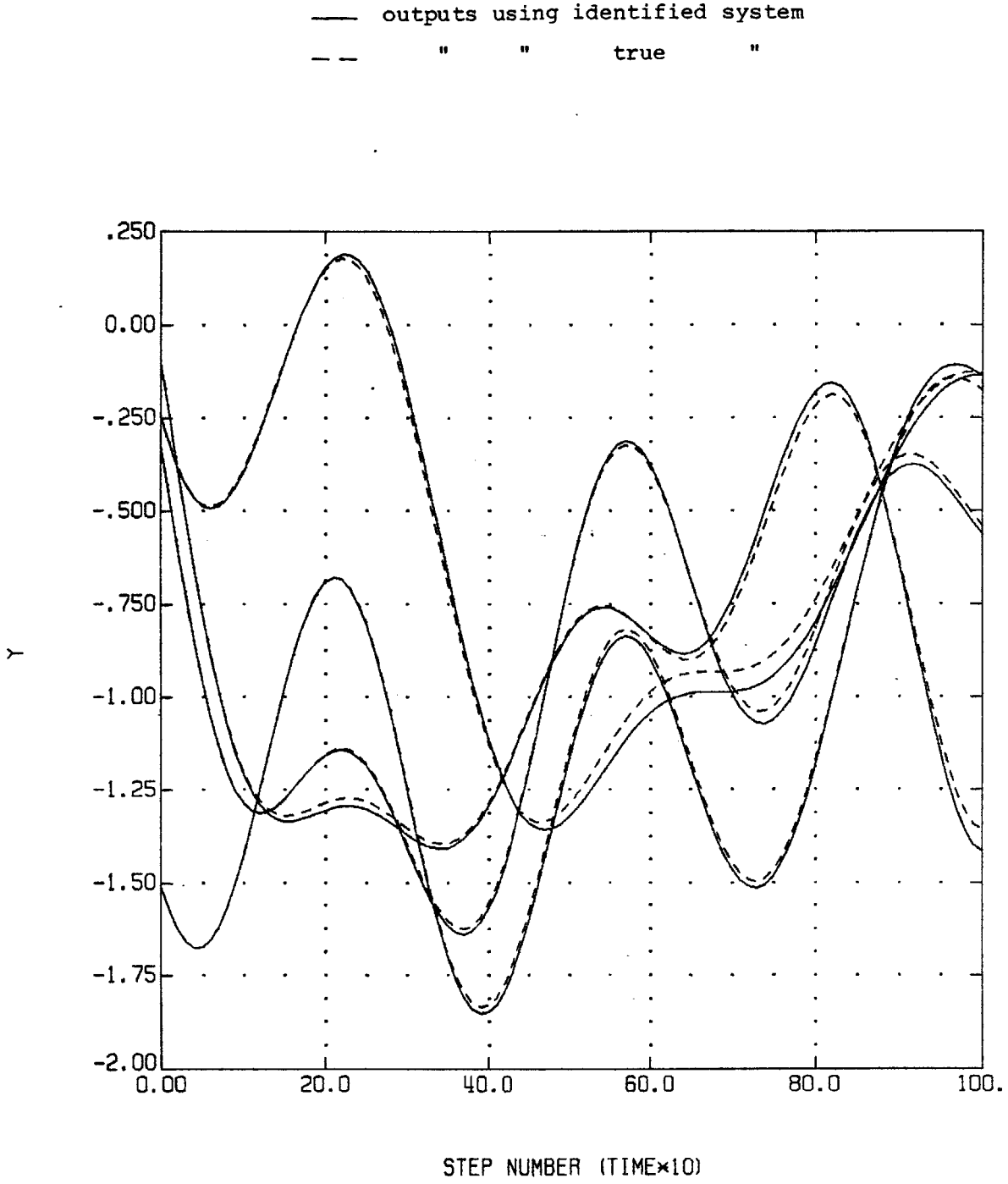


Figure 2.4.11: Outputs for Identified Closed Loop System for 8 Mass Slinky

	Freq.	Damp. Ratio	F/R[1]	Freq.	Damp. Ratio	F/R	Freq.	Damp. Ratio	F/R
True	0.266	.699	R	0.407	.237	R	0.515	.439	F
Identified	0.270	.672		0.408	.226		0.548	.466	
True	0.605	.530	F	0.768	.080	R	0.903	.375	F
Identified	0.581	.555		0.771	.052		0.896	.361	
True	1.112	.028	R	1.168	.145	F	1.385	.107	F
Identified	1.116	.016		1.146	.166		1.418	.012	
True	1.415	.022	R	1.648	.077	F	1.663	.021	R
Identified	1.430	.104		1.661	.016		1.688	.082	
True	1.848	.019	R	1.848	.043	F	1.954	.054	F
Identified	1.847	.018		1.871	.053		1.958	.056	
True	1.962	.022	R						
Identified	1.961	.021							

[1]F/R: F means filter pole, R means regulator pole. For the identified system, the poles don't separate so this cannot be determined directly.

(a) Closed loop poles for steady state filter and regulator

	Modal State Steady State Errors					
	X	V	X	V	X	V
True	.0152	.0036	.0114	.0120	.0095	.0095
Identified	.0152	.0036	.0114	.0120	.0107	.0108
True	.0078	.0078	.0058	.0058	.0040	.0040
Identified	.0101	.0101	.0071	.0072	.0044	.0044
True	.0019	.0019	.0033	.0033		
Identified	.0019	.0019	.0033	.0033		

(b) Standard deviation of steady state errors

Figure 2.4.12: Closed Loop Poles and Steady State Error Standard Deviations for Identified Closed Loop System for 8 Mass Slinky

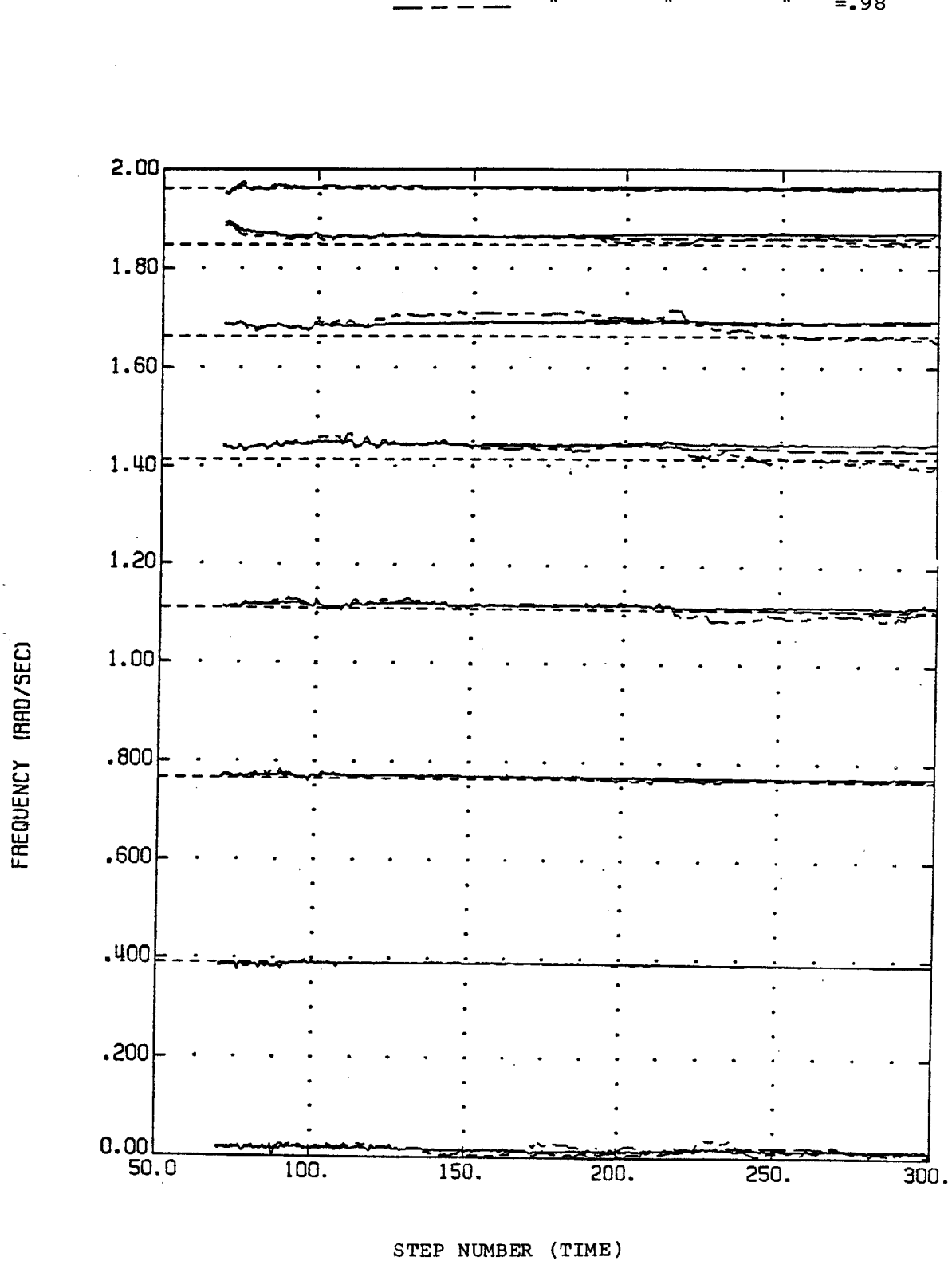


Figure 2.4.13: Comparison of Performance of RLLS and RLELS for 8 Mass Slinky

Chapter 3: Computational Considerations

Chapter 2 described the RLLS algorithm in general. This chapter examines the two computational characteristics which make it particularly attractive for LSSs: cycle time and order recursion. In section 3.1, these are explored for the optimal RLLS algorithm presented in Chapter 2.

As illustrated in Figure 3.1.3, RLLS has a lower cycle time than RLS for the large numbers of identified parameters typical of LSSs. Even so, the RLLS cycle time grows rapidly with the number of parameters. For a minimal identifier (no error checking and no monitoring of the identified modes or choice of system order), the measured cycle time was 15 milliseconds. This limits real-time processing to frequencies lower than about 3 Hz even if the computer is dedicated to identification, the identifier is minimal, only 8 modes are being identified, and the computer is as fast as a VAX 11/782.

Section 3.2 discusses how this cycle time can be reduced with no loss of accuracy by parallel processing. Section 3.3 describes a suboptimal RLLS algorithm which trades off decreased cycle time for increased total number of cycles and/or decreased accuracy.

3.1 Computational Characteristics of RLLS

Cycle time

For RLS, there will be p sets of equations 2.2.1-4, one for each output. Each set will estimate α unknowns. (For a system with N states, m inputs, p outputs, and $\nu_i=N/p$, $\alpha_i = \alpha = \nu_i(p+m) = N(1+m/p)$. If the D matrix were non-zero, α would equal $N(1+m/p) + m$.) The number of calculations required for all p equations is:

Equation	2.2.2 + 2.2.3	2.2.4	Total
Multiplies	$\frac{p}{2}(3\alpha^2+5\alpha)$	$2\alpha p$	$\frac{p}{2}(3\alpha^2+9\alpha)$
Divides	$2\alpha p$	p	$(2\alpha+1)p$
Add/Subtract	$\frac{p}{2}(3\alpha^2+3\alpha-2)$	$2\alpha p$	$\frac{p}{2}(3\alpha^2+7\alpha-2)$

Table 1: Computations Required for UDU^T Version of RLS

The cycle time is limited by the number of multiplies, which goes like $6pN^2$ (for N>>1 and m=p and N evenly divisible by p). RLLS is a transformed version of the equations whose number of multiplies goes like $\frac{4}{3}p^2N$ under those conditions (see Table 2). For N > m,p > 1, as is typical for LSSs, RLLS can cycle more quickly. (NOTE: There is another way to implement RLS where the number of multiplies and additions goes like $8p^2N$. Since this is not as good as RLLS and doesn't have the numerical stability of the UDU^T form, it is not considered in this thesis (ref. 20, p. 340-346.)

The RLLS equations (2.4.4 to 2.4.8) are like 2 RLS estimators plus the calculation of $\bar{e}_n(t)$, $\bar{r}_n(t)$, and $\gamma_n(t)$. The maximum filter order M equals the largest value of ν_i. (For a system with N states, m inputs, p outputs, and ν_i=N/p, α=m+p and M=N/p. If the D matrix were non-zero, $M=\nu_{i_{max}}+1=N/p+1$.) The number of calculations required is:

Equation	Multiplies	Divides	Add/Subtract
2.4.4	$\frac{7}{2}M(\alpha^2+\alpha)$	$M(2\alpha+1)$	$\frac{1}{2}M(7\alpha^2+3\alpha-2)$
2.4.5	$\frac{7}{2}M(\alpha^2+\alpha)$	$M(2\alpha+1)$	$\frac{1}{2}M(7\alpha^2+3\alpha-2)$
2.4.6 + 2.4.7	$2(M-1)\alpha^2$	-	$2(M-1)(\alpha^2+\alpha)$
2.4.8	$\frac{1}{6}(M-1)(\alpha^3+9\alpha^2+2\alpha+6)$	-	$\frac{1}{6}(M-1)(\alpha^3+6\alpha^2+5\alpha+6)$
Total	$\frac{1}{6}(M-1)\alpha^3+\frac{7}{2}(3M-1)\alpha^2+\frac{1}{3}(22M-1)\alpha$	$2M(2\alpha+1)$	$\frac{1}{6}(M-1)\alpha^3+(10M-3)\alpha^2+\frac{1}{6}(35M-17)\alpha-M-1$

Table 2: Computations Required for UDU^T Version of RLLS

The expressions for the number of calculations for equation 2.4.4 (and 2.4.5) are different from those in the Total column in Table 1 because K_n (and K_n^*) are $\alpha\times\alpha$ matrices while $\hat{\theta}_{Ri}$ is a $1\times\alpha$ vector. (Note that even if the expressions were the same, the actual numbers would be different because α in Table 1 is different from α in Table 2.) The number of calculations for equation 2.4.8 is high because the P matrix

is not automatically available when using the UDU^T equations. The α^3 contribution comes from performing U times D times U^T.

Using RLELS requires $\frac{1}{6}\alpha^3 + (M+1)\alpha^2 + (M + \frac{5}{6})\alpha + 1$ multiplications, M divisions, and $\frac{\alpha^3}{6} + (M + \frac{3}{2})\alpha^2 + (M + \frac{1}{3})\alpha + 2p + 1$ add/subtract more than RLLS, and α is increased to m+2p instead of m+p for RLLS.

Figure 3.1.1 is a comparison of the number of multiplies for RLLS, RLELS, and RLS using the expressions from Tables 1 and 2. For these curves, ν_i was assumed to equal N/p and that there is no D matrix. For N (number of states in model) large, both absolutely and relative to p (number of outputs) and m (number of inputs), RLLS is faster than RLS (even though it estimates more parameters). RLELS is substantially slower than RLLS. It is faster than RLS for large systems, although the crossover is at much larger system sizes than for RLLS. Whether this time penalty is worth the gain in accuracy will depend on the application, especially on the amount of noise in the system.

Figure 3.1.2 shows the predicted cycle times for the same cases. The times are for double precision arithmetic on a VAX 11/782 (5.3 μsec for a multiply, 8.5 μsec for a divide, and 4.1 μsec for an add/subtract). (NOTE: These numbers are for double precision operations involving elements of 2 dimensional arrays. The numbers for scalar double precision operations are 3.6, 6.7, and 2.3 respectively.) Comparing these two figures shows that even though cycle time will vary with machine, counting the number of multiplications is a good measure of relative performance of RLLS and RLS.

Table 3 lists the number of calculations and predicted cycle times for systems with 6 or fewer outputs for even larger systems. To minimize the table size, only systems which have no more actuators than sensors are considered. The table also assumes that the number of states is evenly divisible by the number of outputs and that only systems with an even number of states are of interest. For each combination of m and p, two entries are included: either the two values of N where the crossover from RLS being faster to RLLS being faster occurs, or the lowest two values of N. (Note that the assumption of N even means that when p is odd, only every other evenly divisible N is

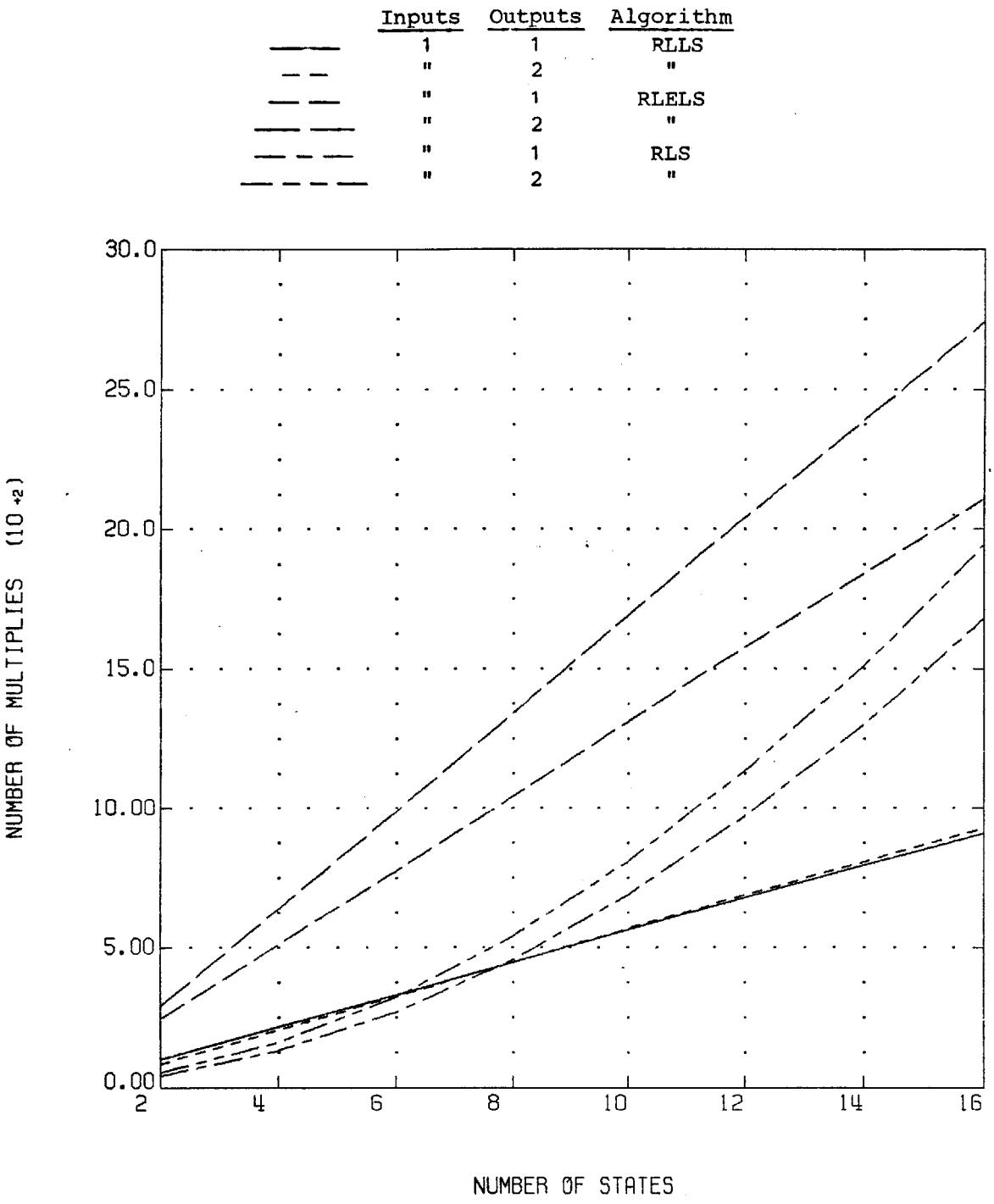


Figure 3.1.1: Comparison of Number of Multiplications for RLLS, RLELS, and RLS

	Inputs	Outputs	Algorithm
———	1	1	RLLS
– –	"	2	"
— – —	"	1	RLS
— – – —	"	2	"

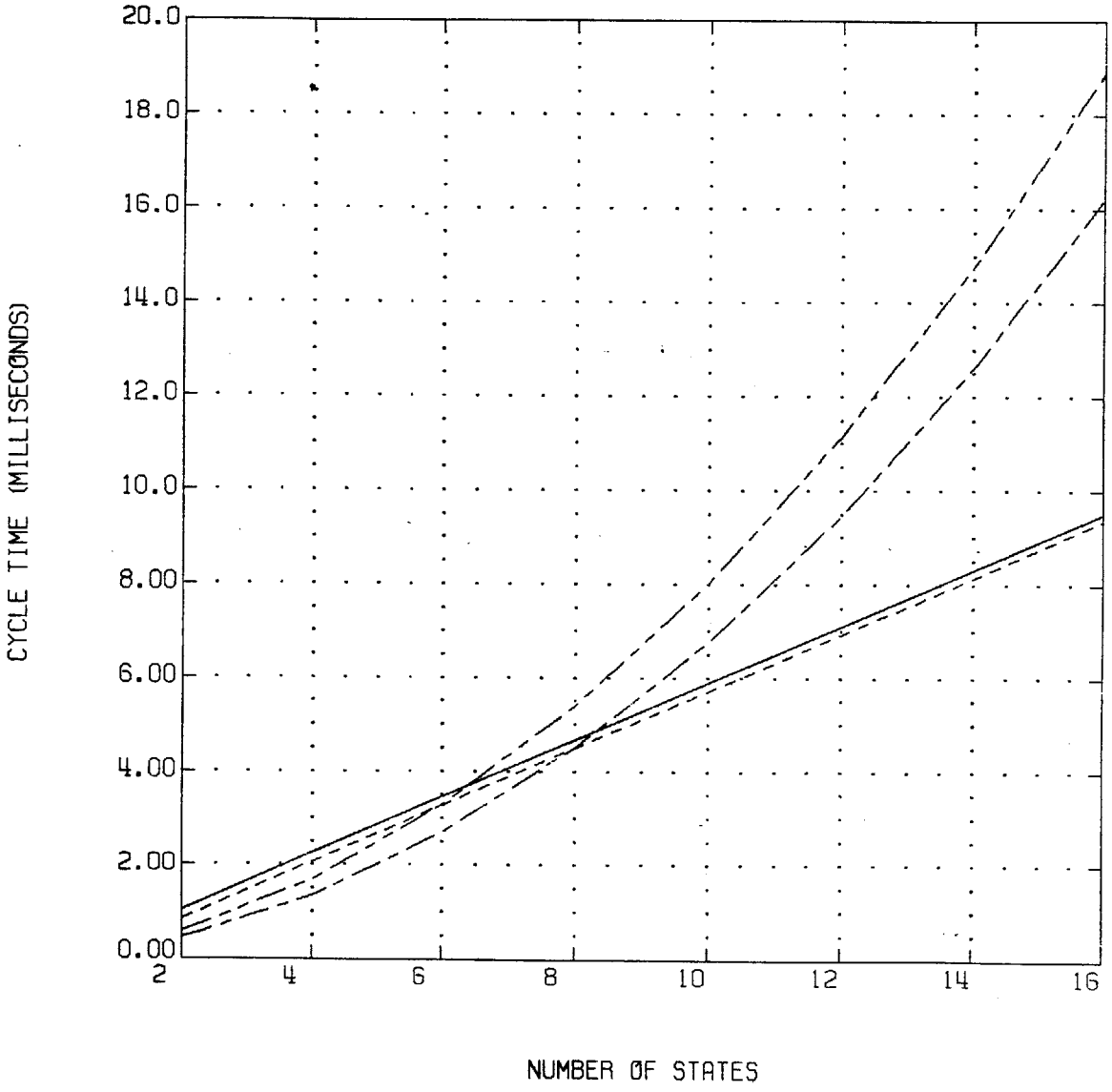


Figure 3.1.2: Predicted Cycle Times for RLLS and RLS

			RLS				RLLS				RLELS
N	m	p	MULT	DIV	ADD	CYCLE (MSEC)	MULT	DIV	ADD	CYCLE (MSEC)	CYCLE (MSEC)
8	1	1	456	33	439	4.50	448	80	396	4.68	10.51
10	1	1	690	41	669	6.75	564	100	500	5.89	13.18
6	1	2	324	38	304	3.29	326	42	292	3.28	9.70
8	1	2	540	50	514	5.39	447	56	403	4.50	13.10
6	2	2	540	50	514	5.39	556	54	508	5.49	13.72
8	2	2	912	66	878	8.99	764	72	701	7.54	18.54
6	1	3	396	51	369	4.04	348	36	315	3.44	12.02
12	1	3	1368	99	1317	13.49	764	72	701	7.54	25.02
6	2	3	585	63	552	5.90	530	44	487	5.18	15.65
12	2	3	2070	123	2007	20.24	1170	88	1085	11.40	32.57
6	3	3	810	75	771	8.09	752	52	698	7.29	19.81
12	3	3	2916	147	2841	28.35	1668	104	1558	16.11	41.23
4	1	4	240	44	216	2.53	210	22	188	2.07	9.11
8	1	4	780	84	736	7.86	530	44	487	5.18	19.82
4	2	4	324	52	296	3.37	294	26	268	2.88	11.28
8	2	4	1080	100	1028	10.79	752	52	698	7.29	24.53
4	3	4	420	60	388	4.33	392	30	362	3.82	13.71
8	3	4	1428	116	1368	14.16	1015	60	949	9.78	29.82
4	4	4	528	68	492	5.39	504	34	470	4.89	16.43
8	4	4	1824	132	1756	17.99	1320	68	1241	12.66	35.70
10	1	5	1350	125	1285	13.49	752	52	698	7.29	29.83
20	1	5	4860	245	4735	47.25	1668	104	1558	16.11	62.04
10	2	5	1785	145	1710	17.70	1015	60	949	9.78	35.71
20	2	5	6510	285	6365	63.02	2261	120	2123	21.71	74.26
10	3	5	2280	165	2195	22.49	1320	68	1241	12.66	42.20
20	3	5	8400	325	8235	81.05	2952	136	2783	28.21	87.73
10	4	5	2835	185	2740	27.83	1668	76	1575	15.94	49.32
20	4	5	10530	365	10345	101.33	3744	152	3541	35.65	102.51
10	5	5	3450	205	3345	33.74	2060	84	1952	19.64	57.09
20	5	5	12900	405	12695	123.86	4640	168	4400	44.06	118.62
6	1	6	630	90	582	6.49	392	30	362	3.82	19.44
12	1	6	2142	174	2052	21.24	1015	60	949	9.78	42.21
6	2	6	792	102	738	8.09	504	34	470	4.89	22.74
12	2	6	2736	198	2634	26.98	1320	68	1241	12.66	49.33
6	3	6	972	114	912	9.86	630	38	592	6.09	26.34
12	3	6	3402	222	3288	33.40	1668	76	1575	15.94	57.10
6	4	6	1170	126	1104	11.80	770	42	728	7.42	30.25
12	4	6	4140	246	4014	40.49	2060	84	1952	19.64	65.54
6	5	6	1386	138	1314	13.91	924	46	878	8.89	34.49
12	5	6	4950	270	4812	48.26	2497	92	2373	23.75	74.66
6	6	6	1620	150	1542	16.18	1092	50	1042	10.48	39.06
12	6	6	5832	294	5682	56.70	2980	100	2839	28.28	84.49

Table 3: Comparison of Predicted Cycle Times for RLLS and RLS

considered.) For p larger than 2, RLLS is faster than RLS starting with the lowest value of N considered, and the discrepancy grows as N increases.

To verify the accuracy of the formulas in Tables 1 and 2, figure 3.1.3 shows predicted cycle times vs. measured cycle times for 1 input, 2 outputs. For the measured times, the program was timed with the identification algorithm subroutine replaced by a null subroutine. This time was subtracted from the runs with the identifier to minimize the amount of overhead time included in the measurements. (Typical values were 6.3 CPU sec for initialization and final print-out and 3.3 CPU millisecond per cycle to generate data and call the subroutines.) The actual performance is about 1.5 times slower than predicted, which is as close as could be expected. These times do not include converting from LLS coefficients to LS coefficients (for RLLS) or any eigenanalysis to check the identified frequencies.

Recursive-by-Order Property

The second characteristic of RLLS (RLELS) which is advantageous for LSSs is choice of model size. Unlike many identification algorithms, RLLS is well behaved if the model size is chosen to be larger than the true system. Thus, if a mode is underexcited so that it disappears from the data, the identifier will not have numerical problems.

Because equations 2.4.4-9 are performed one order at a time, the estimates for low orders are theoretically and in practice unaffected by the estimates for higher orders. "Low orders" refers to the parameter estimates obtained from assuming a low system order. These estimates are corrupted by the presence of unmodelled higher orders and do not correspond to the low order part of the complete system. In other words, K_o (K_o^*) of equation 2.4.4 (2.4.5) is unaffected by the estimate of K_1, $\cdots$, K_{M-1} (K_1^*, $\cdots$, K_{M-1}^*).

This means that the only decision required for RLLS is the maximum identifier order to be considered. If the maximum identifier order is 10 (giving a system order of 10p), then K_8 (K_8^*) is exactly the same

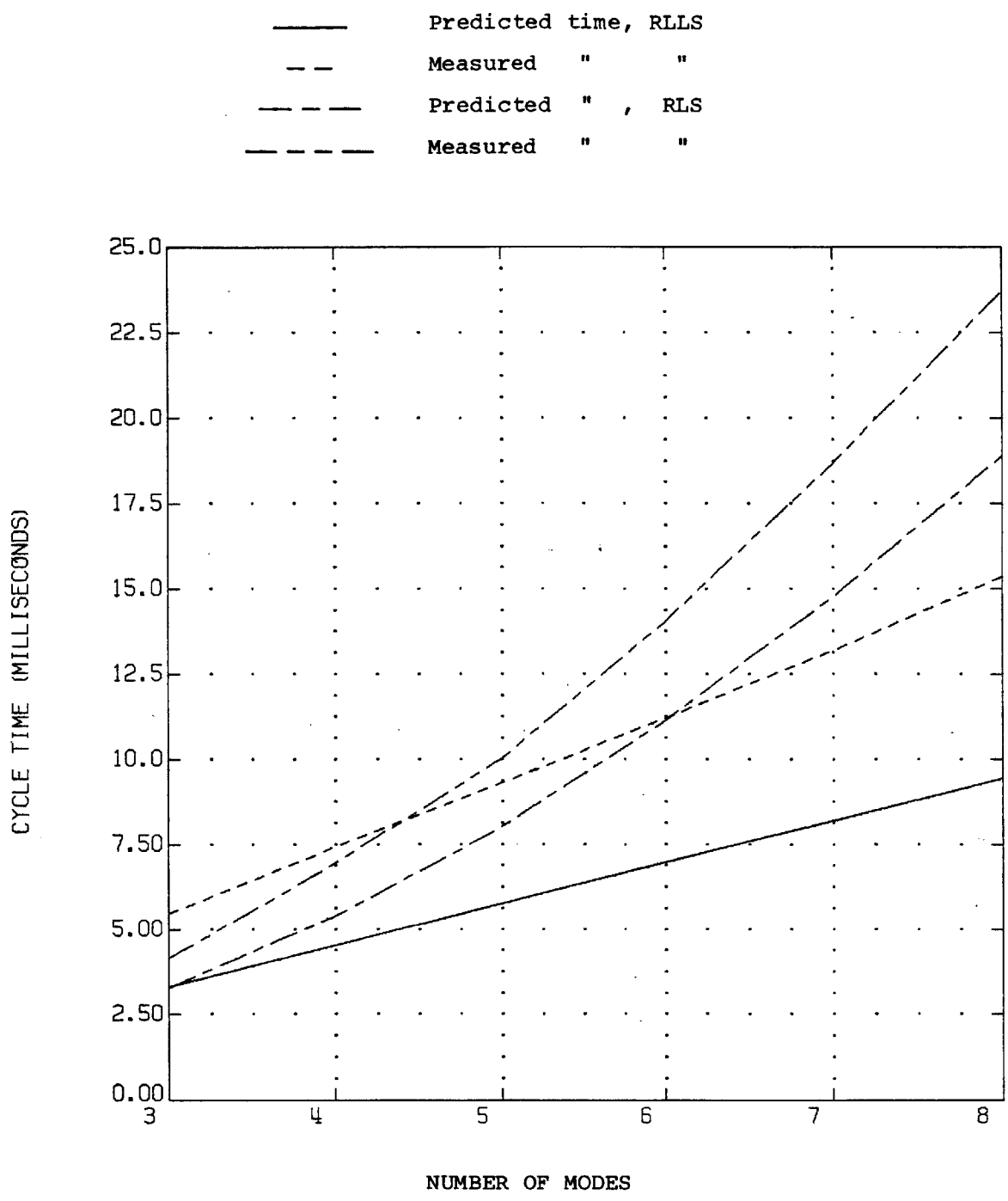


Figure 3.1.3: Predicted vs. Measured Cycle Times
for RLLS and RLS

as if the maximum identifier order had been chosen to be 9. So, the estimated system for all lower identifier orders can be recovered. For RLS, only the estimated system corresponding to the maximum order is recovered. Note that this is not exactly true for RLELS. The estimate of the error $\underline{\varepsilon}(t)$ depends on the maximum identifier order. Since $\underline{\varepsilon}(t)$ affects the estimates of all orders, the lower order systems can still be recovered, but they will vary with the maximum identifier order to some extent.

Figure 3.1.4 shows how this works. In Figure 3.1.4, the dashed lines are the correct frequencies for the 6 modes (there is a rigid body mode at 0 frequency) of a 6 mass slinky. The exact identifier order is also 6 (12 states divided between 2 position sensors, one on mass 1 and one on 5). The assumed maximum identifier order was 9. These results are after 300 steps and a SNR of 85. Spurious modes fitted to the noise appear for orders above 6, but the true modes persist. For low noise levels, this characteristic can be used to identify the system order -- the frequency estimates stabilize abruptly at the correct system order. The noisier the system, the harder this is to judge. For a noise free system, the identifier converges to the true system exactly in a finite number of steps.

Figure 3.1.5 demonstrates this persistence-of-correct-frequency effect, for a 9 mass slinky, even when the true system order changes during an identification. The middle mode (at 1.2856 rad/sec) is deleted at step 100, causing its estimate to drift. Note that it drifts right through the persisting modes with only temporary disturbances to their estimates. (SNR=20 for this graph.)

As demonstrated in the previous pages, RLLS has computational advantages over other algorithms which are important when identifying a LSS specifically. LSS models have so many parameters that the lower cycle time for large models of RLLS is important (see Figure 3.1.2 and Table 3). Also, the identifier is not degraded by choosing a model which is larger than the actual system or by modes which are temporarily underexcited making the system appear smaller than it is (Figure 2.4.16).

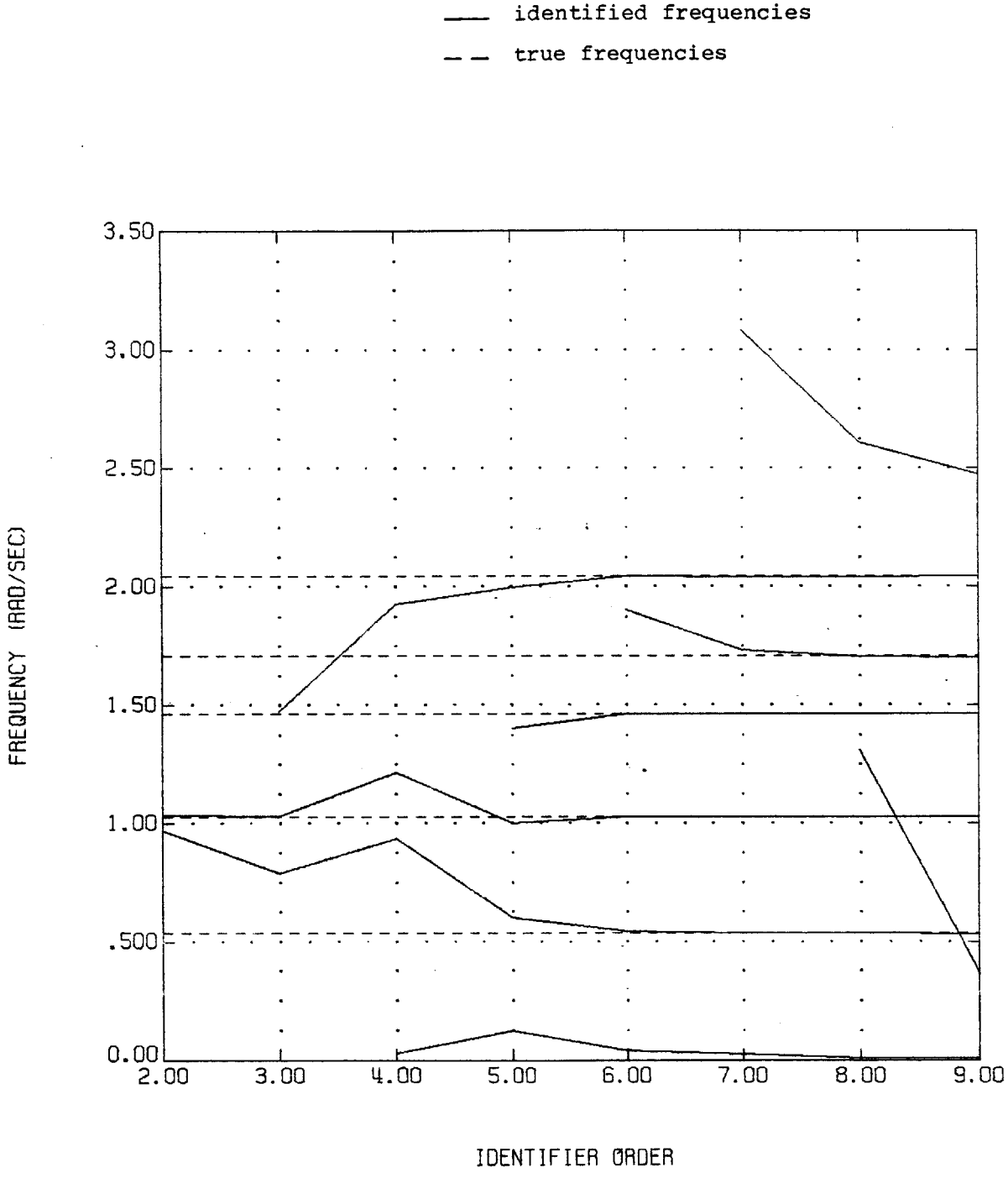


Figure 3.1.4: Identified Frequencies vs. Identifier Order for 6 Mass Slinky

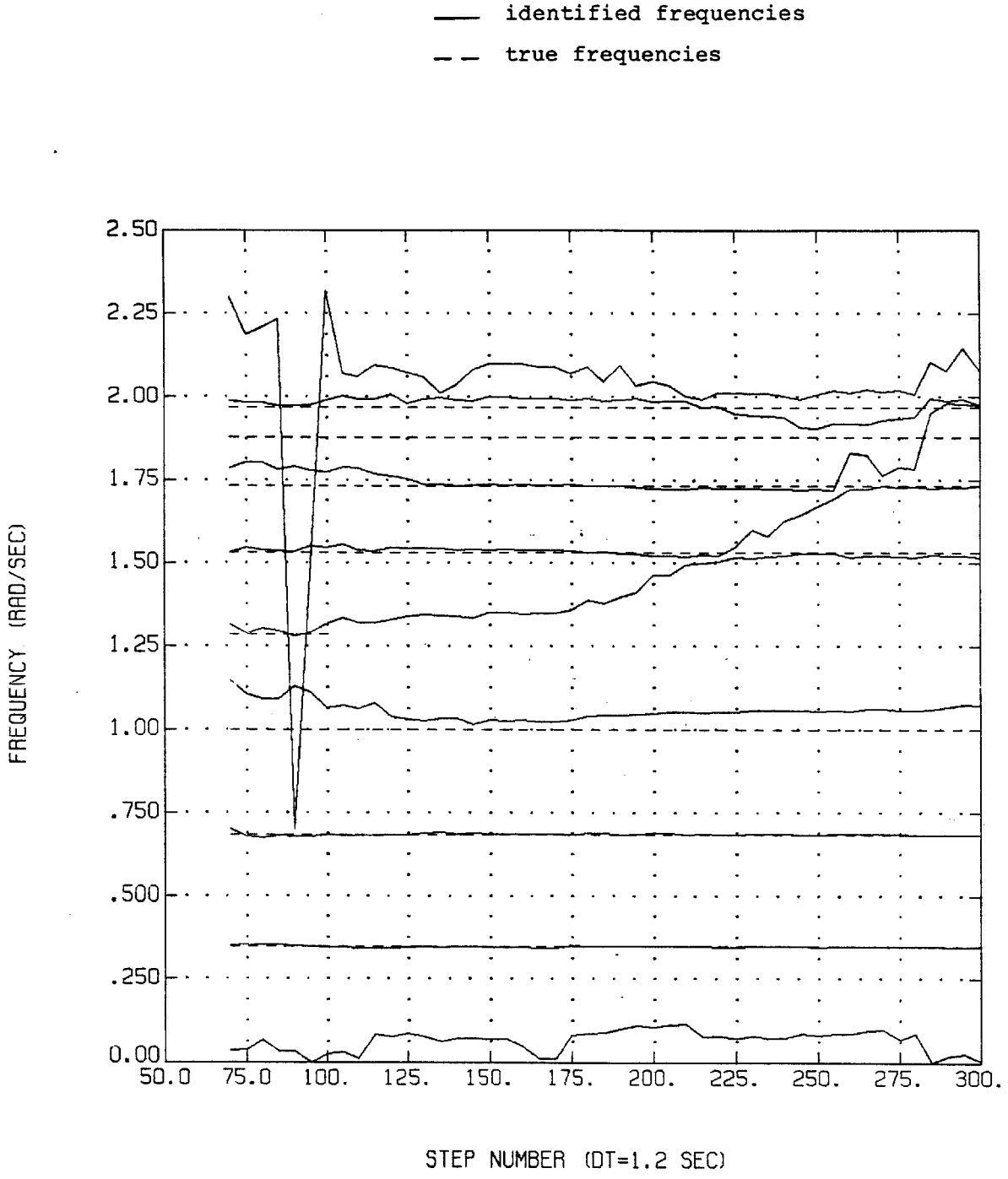


Figure 3.1.5: Identified Frequencies When a Mode Drops Out

Parallel Processing

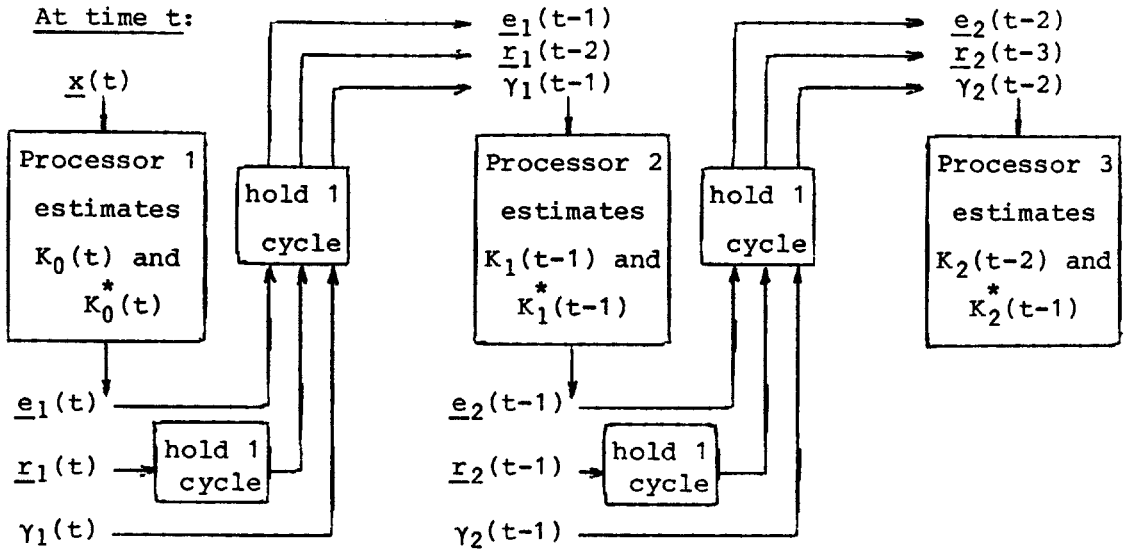
The recursive-by-order property makes RLLS ideally suited to parallel processing. Equations 2.4.4-8 are performed once for each order. Each order needs $\underline{e}(t)$, $\underline{r}(t-1)$ and $\gamma(t)$ from the next lower order, which is a total of $2\alpha+1$ variables. Most of the variables ($\underline{k}$, K, P, $\underline{k}^*$, K^*, and P^*, a total of $4\alpha^2+2\alpha$) are not passed between orders. Thus, the data transmission requirements are not large.

The only change to the algorithm is that the computations would have to be staggered in time. $\underline{e}_{n+1}(t)$ is not available until after $\underline{e}_n(t)$ has been computed, so they can't be calculated in parallel during the same cycle. For example, suppose the identifier is estimating 3 orders, with 3 processors working in parallel. The inputs and outputs are shown in Figure 3.1.6a.

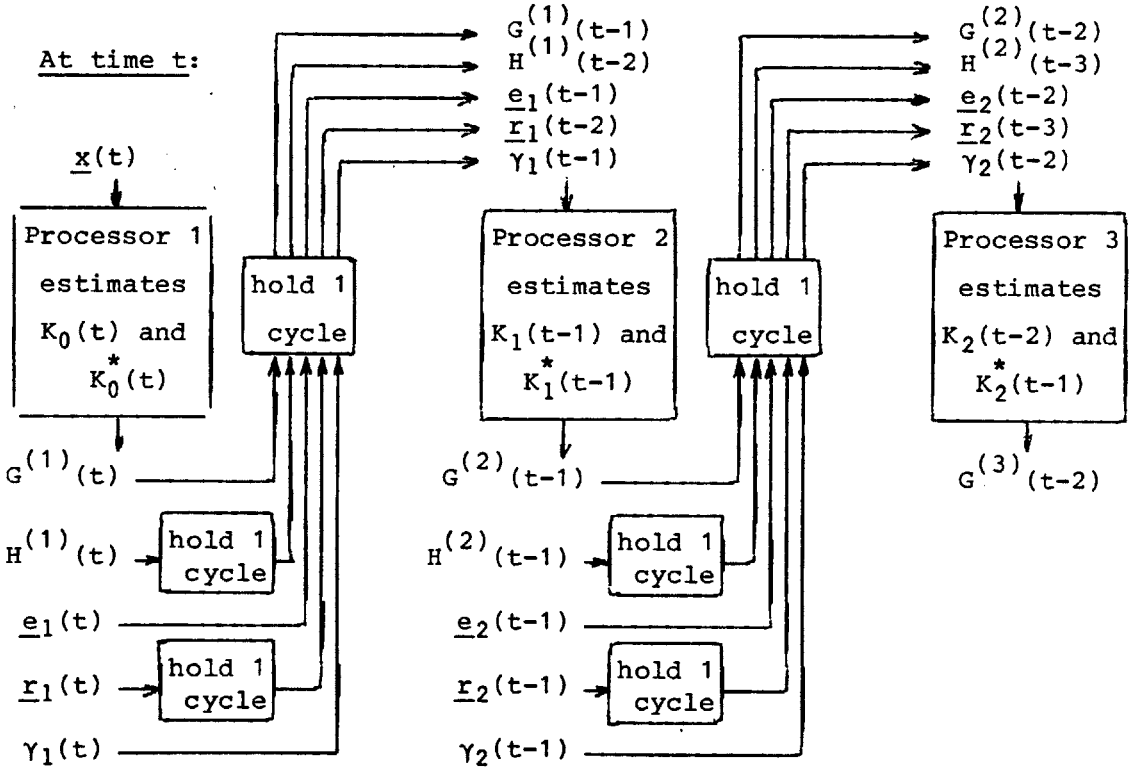
Recovering the parameters in $G^{(3)}(t)$ requires $K_2(t)$, which won't be available until time t+2. The identification will be delayed as many time steps as processors. This is balanced by a decrease in cycle time by a factor which equals the number of processors if the identifier order is divisible by that number; slightly less decrease if not. With this approach, there is no loss of accuracy (unlike the technique discussed in the next section).

Recovering $G^{(n)}(t)$ requires execution of equations 2.4.9, which are not required for the processing shown in Figure 3.1.6a. Either the K and K^* being estimated by each processor needs to be sent to some other processor periodically, or the calculations can be included in the parallel processors, but not executed every cycle. This latter approach is illustrated in Figure 3.1.6b. Note that this requires periodic transmission of G and H between processors, increasing the data transmission requirements.

Again, there is no loss of accuracy but the identifier output is delayed.



(a) No Parameter Recovery



(b) With Parameter Recovery

Figure 3.1.6: RLLS Parallel Processing with Parameter Recovery

3.2 Suboptimal, Low Cycle Time RLLS

The parallel processing technique presented in Section 3.1 has one major drawback--it requires more than one processor. The idea explored in this section is: Suppose you have only 1 processor. How much accuracy do you lose if you approximate parallel processing by (using the example from Figure 3.1.6) running the block of software for Processor 1 for awhile, freezing the values of $K_o(t)$, $K_o^*(t)$, $G^{(1)}(t)$ and $H^{(1)}(t)$, then running the software for processor 2 for awhile, etc.? The cycle time is slightly higher than for parallel processing, because the $\underline{e}(t)$, $\underline{r}(t)$, and $\gamma(t)$ are still generated for the lower orders, but only 1 processor is required and there are no between-processor data transmission requirements. The penalty is in reduced accuracy caused by freezing low orders and in the extended total number of cycles caused by the serial processing. (The number of cycles is extended by the same factor as the cycle time is decreased.)

This procedure is listed in Figure 3.2.1. To see how this differs from the optimal RLLS algorithm, compare it with that given in Figure 2.4.2. As indicated by the parameters LO and HI in Figure 3.2.1, each block of software may estimate any number of identifier orders.

No noise case:

If there is no measurement noise and the noise variance parameter V(t) in the RLLS identifier is set to zero, then the identifier will converge exactly to the true system in a finite number of steps. If the number of cycles per block of identifier orders is greater than that number, the suboptimal identifier will also converge exactly. The reduction in cycle time is acheived with no loss in identifier accuracy.

NOTE: In the case of a RLLS identifier, "exactly" means one of a set of solutions, all of which give the same ARMAX form. The matrices K_o and K_o^* correspond to an identified system which requires only one identifier order. If the real system is bigger than that, there is no single correct value of K_o and K_o^*. Even in the no noise case, their identified values will depend on the input sequence. Every data run

I. Initialization: As in Figure 2.4.2

II. Each new neasurement: $\underline{e}_o(t) = \underline{r}_o(t) = [\underline{y}^T(t)\ \underline{u}^T(t)]^T$

A. For n=0, ..., LO-1:

$$\underline{e}_{n+1}(t) = \underline{e}_n(t) - K_n(t)\ \underline{r}_n(t-1)$$

$$\underline{r}_{n+1}(t) = \underline{r}_n(t-1) - K_n^*(t)\ \underline{e}_n(t)$$

$$\gamma_{n+1}(t) = \gamma_n(t) - \lambda\ \underline{r}_n^T(t-1)\ P_n(t-1)\ \underline{r}_n(t-1)$$

B. For n= LO → HI:

$$\underline{k} = \frac{P_n(t-2)\ \underline{r}_n(t-1)}{\underline{r}_n^T(t-1)\ P_n(t-2)\ \underline{r}_n(t-1) + \gamma_n(t)}$$

$$K_n(t) = K_n(t-1) + [\underline{e}_n(t) - K_n(t-1)\ \underline{r}_n(t-1)]\underline{k}^T$$

$$P_n(t-1) = \frac{1}{\lambda}\{I - \underline{k}\ \underline{r}_n^T(t-1)\}P_n(t-2)$$

$$\underline{k}^* = \frac{P_n^*(t-1)\ \underline{e}_n(t)}{\underline{e}_n^T(t)\ P_n^*(t-1)\ \underline{e}_n(t) + \gamma_n(t)}$$

$$K_n^*(t) = K_n^*(t-1) + [\underline{r}_n(t-1) - K_n^*(t-1)\ \underline{e}_n(t)]\underline{k}^{*T}$$

$$P_n^*(t) = \frac{1}{\lambda}[I - \underline{k}^*\ e_n^T(t)]\ P_n^*(t-1)$$

For n≠HI:

$$\underline{e}_{n+1}(t) = \underline{e}_n(t) - K_n(t)\ \underline{r}_n(t-1)$$

$$\underline{r}_{n+1}(t) = \underline{r}_n(t-1) - K_n^*(t)\ \underline{e}_n(t)$$

$$\gamma_{n+1}(t) = \gamma_n(t) - \lambda\ r_n^T(t-1)\ P_n(t-1)\ r_n(t-1)$$

III. To recover RLS coefficients ($\underline{x}(t) = G(t)\ \underline{\phi}(t-1)$):

For n = LO → HI:

$$G^{(n+1)}(t) = [G^{(n)}(t)\ \ 0] + K_n(t)\ [-H^{(n)}(t-1)\ \ I]$$

$$H^{(n+1)}(t) = [0\ \ H^{(n)}(t-1)] + K_n^*(t)\ [I\ \ -G^{(n)}(t)]$$

where: LO= i*M/NBLOCK
HI = (i+1)*M/NBLOCK - 1
i = largest integer in NCYCLES*NBLOCK/NSTEPS
NCYCLES = number of cycles identifier has executed so far
NBLOCK = number of blocks identifier orders divided into
NSTEPS = total number of cycles the identifier will be run

Figure 3.2.1: Simulated Parallel Processing with RLLS

will produce the same $G^{(n)}$ and $H^{(n)}$ (see equations 2.4.9) for n equal to the true system size, but the Ks and K^*s will vary. Thus, "exactly" means that the true system transfer function is recovered when the assumed system size equals the true system size.

Noise case:

In real systems, there is always noise. The more noise, the more error caused by using the suboptimal identifier. Figures 3.2.2-8 show the results of using the suboptimal identifier on the example examined in Figures 2.4.6-12. The suboptimal identifier required one-quarter as many computations per step as the optimal RLLS identifier.

Comparing the suboptimal results to the optimal results shows: (1) largest frequency error is 4% (vs. 2%); (2) largest damping ratio error is 872% or .14 (vs. 318% or .024); (3) standard deviation of steady state state is .0019 to .025 (vs. .0019 to .015 for the optimal and exact cases).

Note that the maximum decrease in number of computations per step is limited to a factor equal to the maximum identifier order. For SNRs in the range of 100, a factor of four is probably the most which can be expected. The limit on computation decrease factor is not the amount of error introduced by the suboptimality with each decrease in number of computations, but the maximum identifier order which still gives good estimates by the optimal identifier. This in turn depends on the SNR and the difficulty of the identification (spacing of modes, placement of actuators and sensors). For the 8 mass slinky example examined here, the resulting error is about twice that of the optimal identifier, which is still adequate for recovering a useful system model.

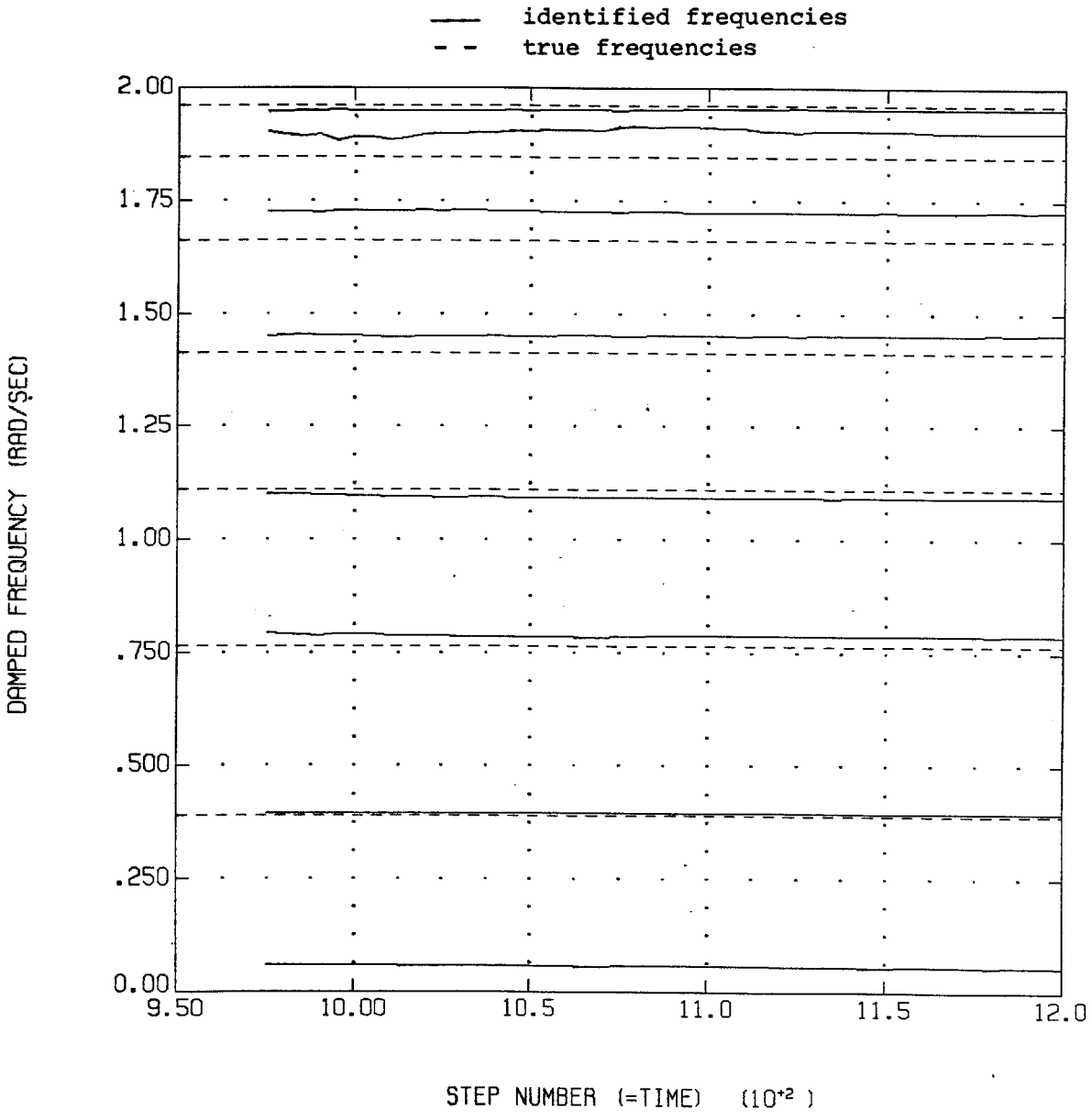


Figure 3.2.2: Suboptimal Identified Frequencies for 8 Mass Slinky

changing lines are identified damping ratios
straight " " true " "

flex mode 1 from Figure 3.2.2
" " 2 " " "
" " 3 " " "
" " 4 " " "
" " 5 " " "
" " 6 " " "
" " 7 " " "

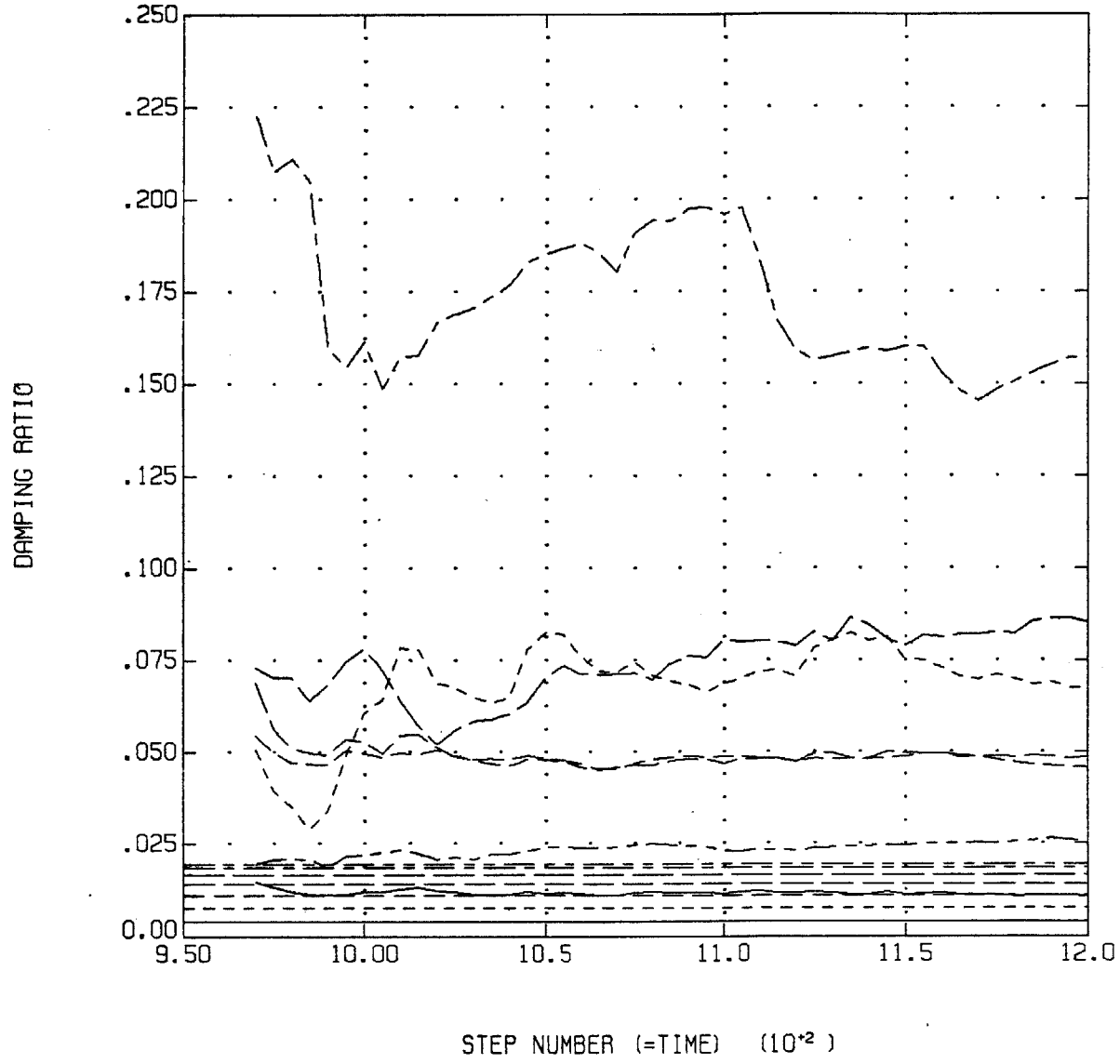


Figure 3.2.3: Suboptimal Identified Damping Ratios for 8 Mass Slinky

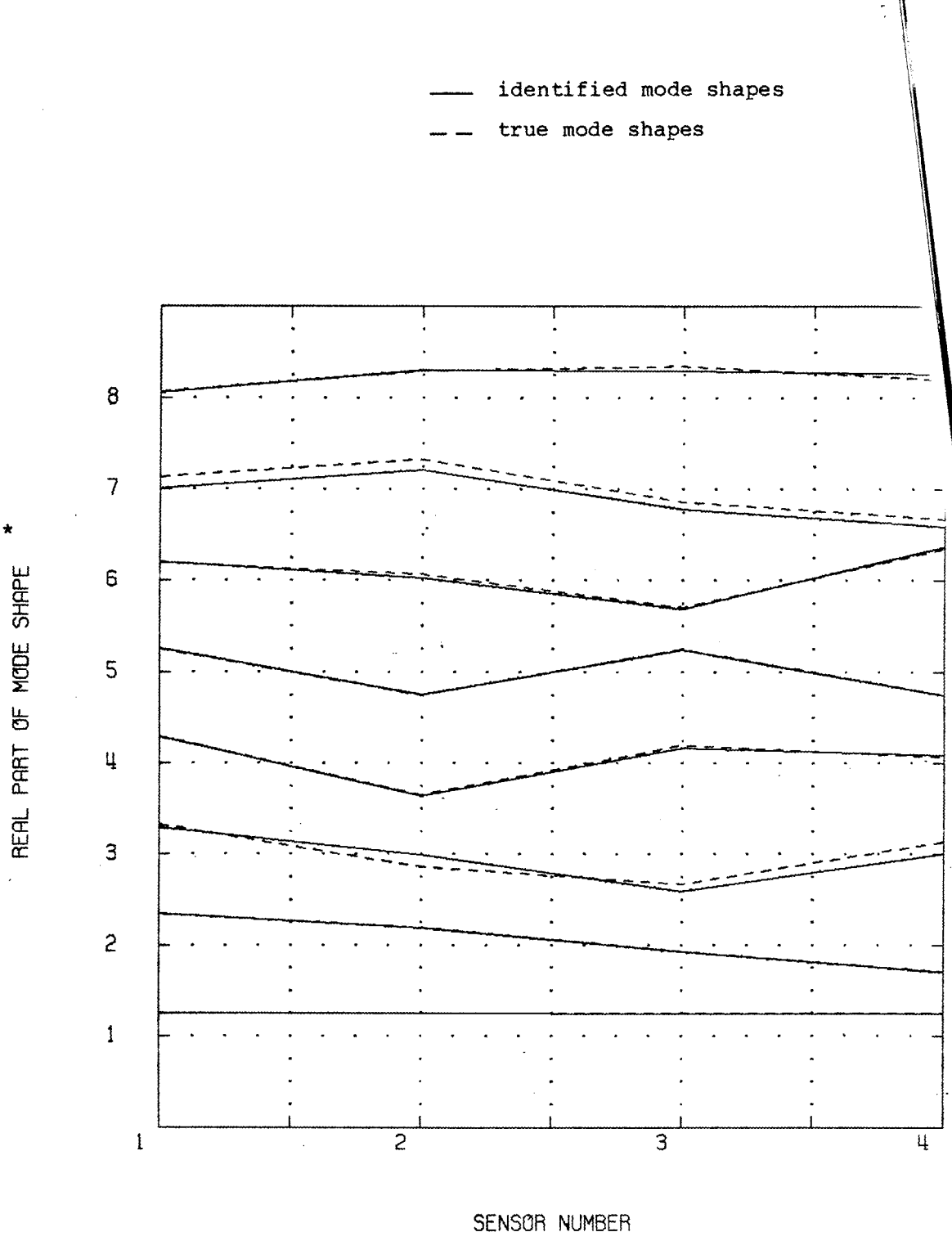


Figure 3.2.4: Suboptimal Identified Mode Shapes for 8 Mass Slinky

*Integers are mode number; deflections about 0 for each integer are the mode shape

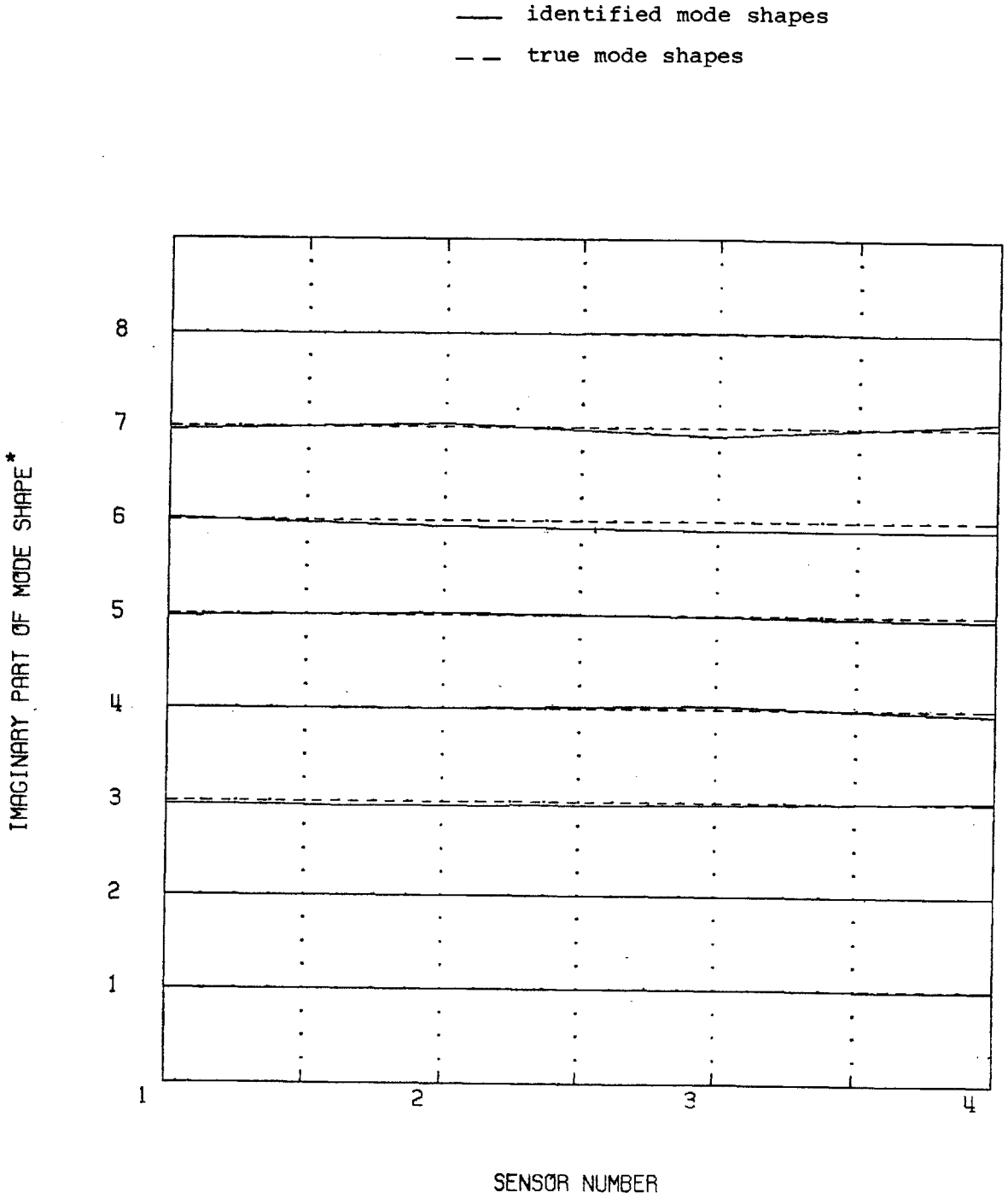


Figure 3.2.5: Imaginary Part of Suboptimal Identified Mode Shapes for 8 Mass Slinky

*Integers are mode number; deflections about 0 for each integer are the mode shape

—— inputs using identified system

– – " " true "

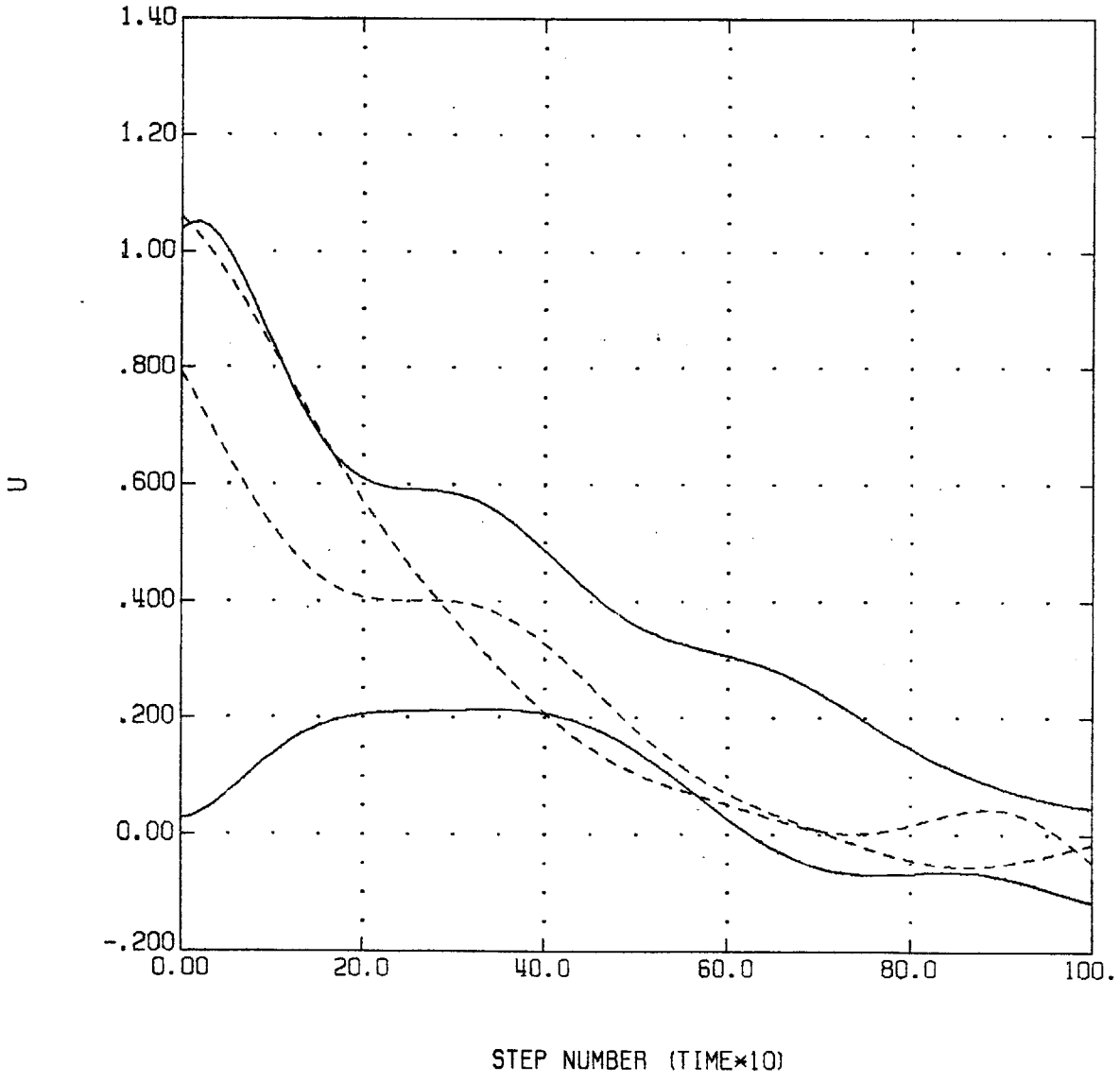


Figure 3.2.6: Control History for Suboptimal Identified Closed Loop System for 8 Mass Slinky

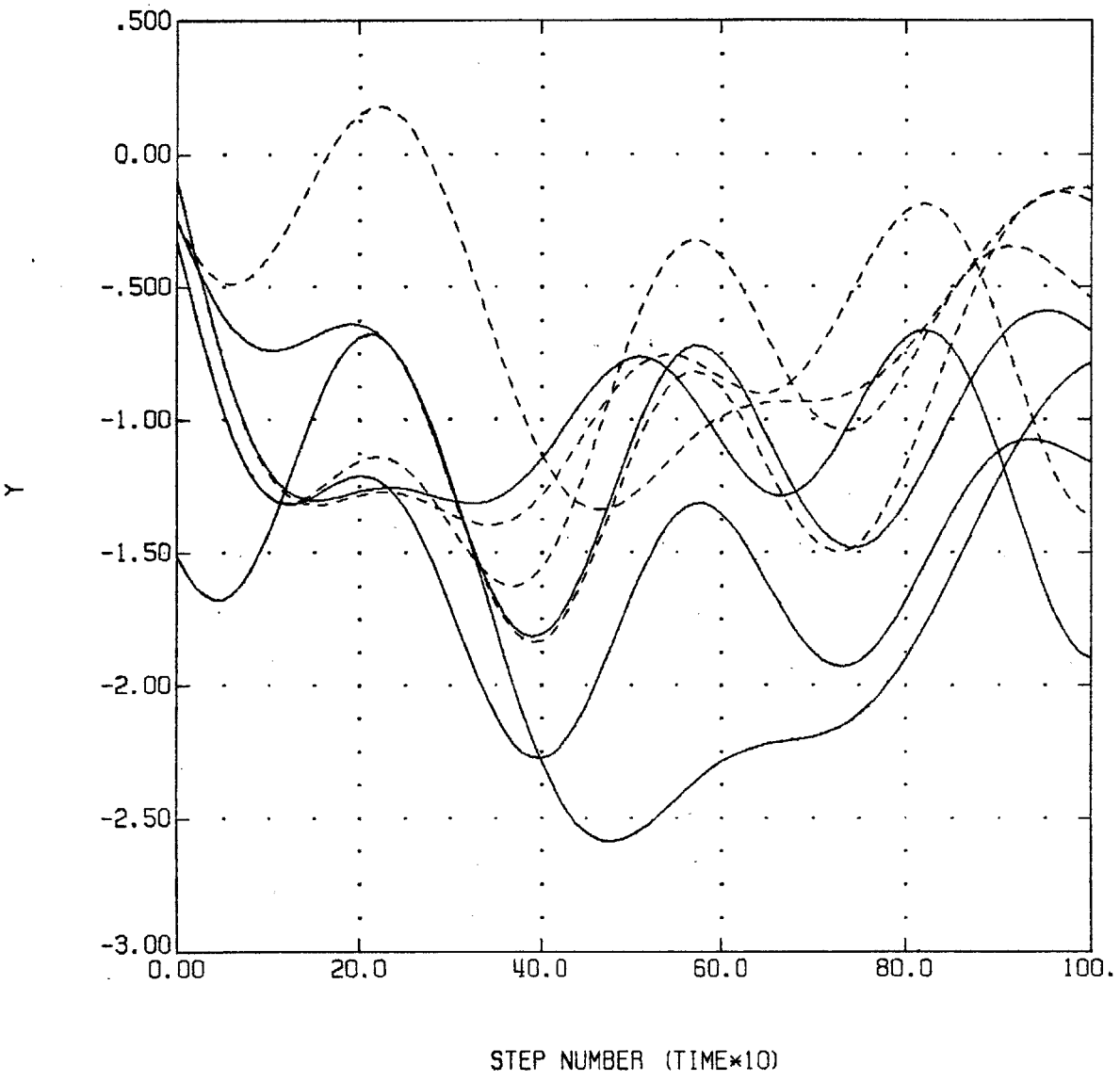


Figure 3.2.7: Outputs for Suboptimal Identified Closed Loop System for 8 Mass Slinky

	Freq.	Damp. Ratio	F/R[1]	Freq.	Damp. Ratio	F/R	Freq.	Damp. Ratio	F/R
True	0.266	.699	R	0.407	.237	R	0.515	.439	F
Identified	0.218	.063		0.424	.116		0.565	.575	
True	0.605	.530	F	0.768	.080	R	0.903	.375	F
Identified	0.780	.038		0.869	.095		1.011	.991	
True	1.112	.028	R	1.168	.145	F	1.385	.107	F
Identified	1.111	.012		1.127	.174		1.414	.020	
True	1.415	.022	R	1.648	.077	F	1.663	.021	R
Identified	1.544	.246		1.652	.014		1.725	.140	
True	1.848	.019	R	1.848	.043	F	1.954	.054	F
Identified	1.853	.017		1.935	.162		1.954	.062	
True	1.962	.022	R						
Identified	1.966	.017							

[1]F/R: F means filter pole, R means regulator pole. For the identified system, the poles don't separate so this cannot be determined directly.

(a) Closed loop poles for steady state filter and regulator

	Modal State Steady State Errors					
	X	V	X	V	X	V
True	.0152	.0036	.0114	.0120	.0095	.0095
Identified	.0252	.0046	.0128	.0139	.0148	.0152
True	.0078	.0078	.0058	.0058	.0040	.0040
Identified	.0098	.0099	.0060	.0059	.0047	.0047
True	.0019	.0019	.0033	.0033		
Identified	.0019	.0019	.0034	.0034		

(b) Standard deviation of steady state errors

Figure 3.2.8: Closed Loop Poles and Steady State Error Standard Deviations for Suboptimal Identified Closed Loop System for 8 Mass Slinky

Chapter 4: Inputs to Enhance Identification

4.1 Introduction

Once the identification algorithm has been chosen, the next step is to choose the input (control) sequence. The necessary and sufficient condition on the input is that it be "persistently exciting and sufficiently rich." A formal definition of these terms can be found in ref. 14, pages 42 and 70. Heuristically, the input must be significantly non-zero during the experiment and must have content at as least as many frequencies as the system being identified. This leaves many choices for an input--step, pulse, pseudo-random binary sequence (PRBS), sum of sinusoids, or some less common function. Some identification algorithms require specific inputs e.g. eigensystem realization requires pulses[16]. Many do not, such as least squares filters.

This chapter presents an algorithm for calculating an optimal input for an unspecified identifier and applies it to an example system. The algorithm was suggested by earlier work by Kalaba & Spingarn[17,18] and Mehra[21], but applying it to LSSs permitted a variation on the approach suggested in their papers which is easier and faster to use. Further, it was extended to consider two problems of particular interest for LSS applications -- very light damping and closely spaced frequencies.

4.2 Figure of Merit

There are two basic theoretical approaches to designing optimal inputs--frequency domain and time domain. Since experiment times for a lightly damped structure are typically much less than the time required to reach sinusoidal steadystate, time domain approaches are more appropriate for LSSs.

The "best" input to use would be that which gives a minimum experimental variance for estimates of the parameters being identified after an experiment of duration T. However, this choice causes the optimal input to depend on the identification algorithm, limiting its usefulness. Also, this would be a complex problem to solve, giving

results which might be too parameter dependent to apply in a real situation, and providing little insight.

To avoid the problem of identification algorithm dependency, select J to be the sensitivity of the output to the parameter(s) of interest. Maximizing the sensitivity should make a parameter easier to determine. Since estimate variances at T depend on all the data from time 0 to T, choose:

$$J = \frac{1}{2} \int_0^T \left(\sum_{i=1}^{r} k_i \, \underline{y}_{\alpha_i}^T \, \underline{y}_{\alpha_i} \right) dt \qquad (4.2.1)$$

where:

$\underline{y}$ = output

$\underline{y}_{\alpha_i}$ = partial derivative of $\underline{y}$ with respect to α_i

α_i = parameter of interest

k_i = relative weighting when several parameters are being estimated

This choice for J has a more theoretical justification than that given above. If the k_i equalled the inverse of the variance of the measurement noise, J would equal the trace of the information matrix[21]. The diagonal elements of the inverse of the information matrix are the Cramer-Rao lower bounds on the variances of the estimates. The inverse of the trace of the information matrix and the trace of the inverse have similar asymptotic behavior[1], so maximizing J should decrease the variance of the estimates.

The J of equation 4.2.1 cannot be used to calculate optimal inputs because it grows as the magnitude of the input grows, giving an answer of input=∞. Some constraint must be put on the input to get a usable answer. One of three constraints is most likely to appear in practice:

1. On-off input i.e. $|u|$ = fixed
2. Available input is limited i.e. $|u| <=_T$ fixed
3. Input usage needs to be minimized i.e. $\int_0^T u^2 \, dt <=$ fixed

Flexible LSSs are almost certain to have proportional actuators for good control of the flexible motion. These actuators will be

limited in the maximum amount of torque/force they can provide. This makes choice 2 the most realistic. Using that in an optimal control algorithm is not only difficult but will probably produce bang-bang trajectories, as if you had selected choice 1. These may be difficult to implement on a proportional actuator.

Because of the difficulty of using constraint 2, the initial investigation used constraint 3. Since this gave good results, the constraint 2 case was not carefully studied and might be a good area for future work.

So, the problem statement was changed to:

$$J = \frac{1}{2}\int_0^T \left(\sum_{i=1}^{r} k_i \, \underline{y}_{\alpha_i}^T \underline{y}_{\alpha_i}\right) dt \qquad (4.2.2)$$

$$\text{subject to: } \frac{1}{2}\int_0^T \underline{u}^T \underline{u} \, dt = E \qquad (4.2.3)$$

Note that constraint (4.2.3) forces the input to be non-zero i.e. "do nothing" is not an allowable solution.

4.3 Necessary (First Order) Conditions

The possible solutions are given by applying the Pontryagin maximum principle. For a LSS, the parameters of interest are structural frequencies, damping ratios, and mode shapes, so the state equations are written in modal form:

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ -\omega_0^2 & -2\zeta_0\omega_0 & & & \vdots \\ & & & & \vdots \\ 0 & & \ddots & & \vdots \\ \vdots & & & & 0 \\ \vdots & & & & \\ & & & 0 & 1 \\ 0 & \cdots & 0 & -\omega_{n-1}^2 & -2\zeta_{n-1}\omega_{n-1} \end{bmatrix} \underline{x} + \begin{bmatrix} 0 & \cdots & 0 \\ b_{2,1} & & \vdots \\ 0 & & \vdots \\ b_{4,1} & & \vdots \\ \vdots & & \\ 0 & & 0 \\ b_{2n,1} & \cdots & b_{2n,m} \end{bmatrix} \underline{u} \qquad (4.3.1)$$

$$\underline{y} = \begin{bmatrix} c_{1,1} & \cdots & c_{1,2n} \\ \vdots & & \vdots \\ c_{p,1} & \cdots & c_{p,2n} \end{bmatrix} \underline{x} \tag{4.3.2}$$

or:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \tag{4.3.3}$$

$$\underline{y} = C\underline{x} \tag{4.3.4}$$

The α_i are $\omega_0, \ldots, \omega_{n-1}\zeta_0, \ldots, \zeta_{n-1}b_{2,1}, \ldots, b_{2n,m}$. The case of unknown elements in C will be discussed in section 4.8. Defining the augmented state $\underline{\xi} = \begin{bmatrix} \underline{x} \\ \underline{x}_{\alpha_1} \\ \vdots \\ \underline{x}_{\alpha_r} \end{bmatrix}$, the state equations become:

$$\dot{\underline{\xi}} = \begin{bmatrix} A & & & \\ A_{\alpha_1} & A & & 0 \\ A_{\alpha_2} & 0 & A & \\ \vdots & & & \ddots \\ A_{\alpha_r} & \cdots & & A \end{bmatrix} \underline{\xi} + \begin{bmatrix} B \\ B_{\alpha_1} \\ \vdots \\ B_{\alpha_r} \end{bmatrix} \underline{u} \tag{4.3.5}$$

$$\underline{y}_{\alpha_i} = C\,\underline{x}_{\alpha_i} \quad \text{i.e.} \begin{bmatrix} \underline{y}_{\alpha_1} \\ \vdots \\ \underline{y}_{\alpha_r} \end{bmatrix} = \begin{bmatrix} 0 & C & & \\ \vdots & & C & 0 \\ 0 & 0 & & C \end{bmatrix} \underline{\xi}$$

or:

$$\dot{\underline{\xi}} = F\underline{\xi} + G\underline{u} \tag{4.3.6}$$

$$[\underline{y}_{\alpha_i}] = H\underline{\xi} \tag{4.3.7}$$

Define an additional state x_o and weighting matrix K, where:

$$\dot{x}_o = \underline{u}^T\underline{u} \qquad \text{with} \quad x_o(0) = 0,\ x_o(T) = 2E$$

$$K = \begin{bmatrix} k_1 I_p & & 0 \\ & \ddots & \\ 0 & & k_r I_p \end{bmatrix} \qquad \text{with} \quad I_p \triangleq \text{pxp identity matrix}$$

The boundary conditions on $\underline{\xi}$ are the initial conditions on $\underline{x}$ and $\underline{x}_{\alpha_i}(0)=\underline{0}$ (since $\underline{x}(0)$ is fixed). The terminal state is free. Collecting these equations, the problem is:

$$\text{Minimize over u:} \quad J = -\frac{1}{2}\int_0^T (\underline{\xi}^T H^T K H \underline{\xi})\, dt \tag{4.3.8}$$

$$\text{Subject to:} \quad \dot{\underline{\xi}} = F\underline{\xi} + G\underline{u} \tag{4.3.9}$$

$$\dot{x}_o = \underline{u}^T\underline{u} \tag{4.3.10}$$

$$\text{B.C.s:} \quad \underline{\xi}(0) = \begin{bmatrix} \underline{x}(0) \\ \underline{0} \end{bmatrix} \tag{4.3.11}$$

$$x_o(0) = 0 \tag{4.3.12}$$

$$x_o(T) = 2E \tag{4.3.13}$$

Solving this:

$$\text{Hamiltonian} \triangleq \Gamma = -\frac{1}{2}\underline{\xi}^T H^T K H \underline{\xi} + \underline{\lambda}^T(F\underline{\xi} + G\underline{u}) + \lambda_o \underline{u}^T\underline{u} \tag{4.3.14}$$

$$\dot{\underline{\lambda}} = -\frac{\partial \Gamma}{\partial \underline{\xi}}^T = H^T K H \underline{\xi} - F^T\underline{\lambda}$$

$$\dot{\lambda}_o = -\frac{\partial \Gamma}{\partial x_o} = 0$$

$$\frac{\partial \Gamma}{\partial u} = 0 = \underline{\lambda}^T G + 2\lambda_o \underline{u} \ ; \ \underline{u} = -\frac{1}{2\lambda_0} G^T\underline{\lambda}$$

$$\underline{\lambda}(T) = \underline{0}$$

$$\lambda_o(T) = \nu$$

Solving the λ_o equations:

$$\dot{\lambda}_o = 0 \text{ and } \lambda_o(T) = \nu \rightarrow \lambda_o(t) = \nu$$

Combining these equations, the solution is:

$$\begin{bmatrix} \dot{\underline{\xi}} \\ \dot{\underline{\lambda}} \end{bmatrix} = \begin{bmatrix} F & -\frac{1}{2\nu} GG^T \\ H^T K H & -F^T \end{bmatrix} \begin{bmatrix} \underline{\xi} \\ \underline{\lambda} \end{bmatrix} \triangleq F' \begin{bmatrix} \underline{\xi} \\ \underline{\lambda} \end{bmatrix}$$

$$\text{with } \underline{\xi}(0) = \begin{bmatrix} \underline{x}(0) \\ \underline{0} \end{bmatrix} \tag{4.3.15}$$

$$\underline{\lambda}(T) = \underline{0}$$

$$\underline{u} = -\frac{1}{2\nu} G^T \underline{\lambda} \tag{4.3.16}$$

$$\frac{1}{4\nu^2} \int_0^T \underline{\lambda}^T GG^T \underline{\lambda} dt = 2E \tag{4.3.17}$$

Two solutions to this two point boundary value problem were proposed: one by Kalaba & Spingarn[17,18] and one by Mehra[21]. The approach in ref. 17 and 18 works only for $\underline{x}(0) \neq \underline{0}$; that in ref. 21 only for $\underline{x}(0) = \underline{0}$. Both require numerical integration. The numerical integration limits the accuracy of the solution and requires long computer runs for experiment times longer than a few seconds.

This thesis uses a third approach which takes advantage of the uncoupled modes model of a LSS. This approach replaces the numerical integration with an eigensolution and will work for either $\underline{x}(0) = \underline{0}$ or $\underline{x}(0) \neq \underline{0}$. Assume a solution of the form:

$$\begin{bmatrix} \underline{\xi}(t) \\ \underline{\lambda}(t) \end{bmatrix} = \sum_{i=1}^{4n(r+1)} \gamma_i \underline{\phi}_i e^{\beta_i t} \triangleq \Phi \underline{\gamma} \tag{4.3.18}$$

where: β_i = eigenvalues of F' (see equation 4.3.15)

$\underline{\phi_i}$ = normalized eigenvectors of F'

γ_i = constant chosen to meet the boundary conditions of equation 4.3.15

Φ = matrix whose columns are $\underline{\phi_i}\, e^{\beta_i t}$

$\underline{\gamma}$ = column vector whose elements are γ_i.

Solving for Φ and $\underline{\gamma}$ as stated involves inverting a rather large matrix and computing independent eigenvectors for repeated eigenvalues. Solutions to these two problems are discussed in section 4.10. The procedure for finding Φ and $\underline{\gamma}$ if the above were easy to do is:

1) Choose ν (see step 4 and comment 2). Construct F' (equation 4.3.15).
2) Find the eigenvalues and eigenvectors of F'.
3) Evaluate the rows of Φ (eq. 4.3.18) corresponding to elements of $\underline{\xi}$ (i.e. rows 1 to 2n × (r+1)) at t=0. Evaluate the rest of the rows at t=T. Call this matrix Φ_c.
4) If $\underline{x}(0)\neq\underline{0}$, $\underline{\gamma}=\Phi_c^{-1}\left[\underline{\xi}^T(0)\ \ \underline{\lambda}^T(T)\right]^T=\Phi_c^{-1}\left[\underline{\xi}^T(0)\ \ \underline{0}^T\right]^T$. If $\underline{x}(0)=\underline{0}$, ν was chosen so that $\left|\Phi_c\right|=0$. Solve for the elements of $\underline{\gamma}$ in terms of one of its elements. The amplitude of that element is chosen to satisfy equation 4.3.17.

Comment 1: As described in step (4), the algorithm is affected by whether or not $\underline{x}(0)=\underline{0}$, but the variation is a minor part of the algorithm.

Comment 2: To get an optimal input for an experiment where T is specified will require iterating through steps 1-4. Pick a value of ν and: iterate on the value of T until $\left|\Phi_c\right|=0$ if $\underline{x}(0)=\underline{0}$; or (2) evaluate equation 4.3.17 if $\underline{x}(0)\neq\underline{0}$. If T ($\underline{x}(0)=\underline{0}$) or E ($\underline{x}(0)\neq\underline{0}$) is wrong, try a new value of ν and iterate. The system is well-behaved, so that a search algorithm can be used to make the process very fast. A similar iteration sequence is required by the approaches of ref. 17

and 21. The method proposed here is faster and more accurate than their approaches because the numerical integration is eliminated. (For example, ref. 18 gives a numerical example of a SISO single state system. The reported value of T for the given ν is 1.02 while the correct value to three decimal places is 1.012.) In addition (as discussed in section 4.10), using the modal form uncouples some of the variables and they can be removed from the calculations for ν and T. This reduces the number of variables used, saving even more time.

4.4 Sufficient (Second Order) Conditions

The sufficient conditions for a weak local minimum of the system given by equations 4.3.14, 4.3.15, and 4.3.16 are (ref. (E) for the case $\underline{x}(T)$=free):

$$\Gamma_{uu}(t) > 0 \quad \text{for } 0 \leq t \leq T \quad \text{(strengthened Legendre-Clebsch)} \tag{4.4.1}$$

$$S(t) \text{ finite} \quad 0 \leq t < T \text{ (Jacobi or no conjugate point condition)} \tag{4.4.2}$$

where: $$\dot{S} = -SF - F^T S + \frac{1}{2\nu} SGG^T S + H^T K H \quad \text{with } S(T)=0 \tag{4.4.3}$$

Differentiating equation 4.3.14 twice gives:

$$\Gamma_{uu} = 2\lambda_o I = 2\nu I$$

Since $\nu>0$ always (see Section 4.6, "Characteristics of the Solution", comment 4) equation 4.4.1 is satisfied. To check equation 4.4.2, consider the Ricatti equation form of the solution to equation 4.3.15 (ref. (D)):

$$\underline{\lambda}(t) = S(t)\underline{\xi}(t) \tag{4.4.4}$$

$$\dot{S} = -SF - F^T S + \frac{1}{2\nu} SGG^T S + H^T K H \quad \text{with } S(T)=0 \tag{4.4.5}$$

Note that equations 4.4.5 and 4.4.3 are the same. From equations 4.4.4 and 4.4.5, if S goes to infinity in the interval $0 \leq t < T$, then $\underline{\lambda}(t)$ also goes to infinity for $\underline{\xi}(t) \neq \underline{0}$. Thus, the condition on the equation for a solution to exist (4.4.5) and the condition (equation 4.4.2) on the

equation for there to be no conjugate point (4.4.3) are the same. In addtion, for $\underline{x}(0)\neq\underline{0}$, there can be no conjugate point at T, because then S(0) is infinite, $\underline{\xi}(0)$ is non-zero, giving $\underline{\lambda}(0)$ infinite. For $\underline{x}(0)=\underline{0}$, the solution when there are no conjugate points in $0\leq t\leq T$ is $\underline{\lambda}(t) = \underline{\xi}(t) = \underline{0}$. This is a valid solution in that it is an optimum, but it is not useful. If there is a conjugate point at T, then there exists a nontrivial perturbation to the optimum which satisfies the first order conditions and is zero at 0 and T[8]. Thus, the boundary conditions $\underline{\xi}(0)=\underline{0}$ and $\underline{\lambda}(T)=\underline{0}$ can be met with a nontrivial $\underline{\xi}(t)$ and $\underline{\lambda}(t)$ only at a conjugate point. As shown in step (4) at the end of section 4.3, this means there is a conjugate point at T such that $\left|\Phi_c\right| = 0$:

> For $\underline{x}(0)\neq\underline{0}$, the algorithm provides a weak local minimum provided T < first conjugate point. For $\underline{x}(0)=\underline{0}$, the algorithm has no solution except at conjugate points, and is minimum only if T = first conjugate point.

Because of the form of equation 4.3.18, $\underline{\xi}(t)$ and $\underline{\lambda}(t)$ are sums of exponentials, some of which are oscillatory, so that $\left|\Phi_c\right|=0$ for an infinite number of periodically repeating T, given a particular system and choice of ν and E. Past the first such T, numerical results show that the solution is maximizing, although there is no theory that allows determination of optimality for times larger than the first conjugate point.

4.5 Simple Example

A low order example may clarify this procedure. Consider the case used in ref. 18, i.e. SISO single state with (single element) state matrix unknown. In the notation of eq. 4.3.1:

$$\dot{x} = ax + bu$$
$$y = cx + \text{noise}$$

Assume b=c=1 and a is unknown but the true value is -.1. The weighting parameter $k_1=1$. Following the procedure from section 4.3:

1. Choose ν. Construct F'.

Choose $\nu = .0375$. F' becomes:

$$F' = \begin{bmatrix} F & -\frac{1}{2\nu} GG^T \\ H^T K H & -F^T \end{bmatrix} = \begin{bmatrix} A & 0 & -\frac{1}{2\nu} BB^T & 0 \\ A_a & A & 0 & 0 \\ 0 & 0 & -A^T & -A_a^T \\ 0 & C^T K C & 0 & -A^T \end{bmatrix}$$

$$= \begin{bmatrix} a & 0 & -\frac{b^2}{2\nu} & 0 \\ 1 & a & 0 & 0 \\ 0 & 0 & -a & -1 \\ 0 & k_1 c^2 & 0 & -a \end{bmatrix} = \begin{bmatrix} -.1 & 0 & -13.3 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & .1 & -1 \\ 0 & 1 & 0 & .1 \end{bmatrix}$$

2. Find the eigenvalues and eigenvectors of F'.

$$\left| F' - \beta I \right| = 0 \rightarrow \beta = \pm \sqrt{a^2 + bc \sqrt{\frac{k_1}{2\nu}}}, \pm \sqrt{a^2 - bc \sqrt{\frac{k_1}{2\nu}}}$$

$$= \pm 1.9135, \pm 1.9083\, i$$

$$\underline{\phi}_i = \begin{bmatrix} 1 \\ \frac{1}{\beta_i - a} \\ \frac{2\nu}{b^2}(a - \beta_i) \\ \frac{2\nu}{b^2}(\beta_i^2 - a^2) \end{bmatrix}$$

$$\Phi = \begin{bmatrix} e^{1.914t} & e^{-1.914t} & e^{1.908it} & e^{-1.908it} \\ .50e^{1.914t} & -.55e^{-1.914t} & (.027-.52i)e^{1.908it} & (.027+.52i)e^{-1.908it} \\ -.15e^{1.914t} & .14e^{-1.914t} & (-.008-.14i)e^{1.908it} & (-.008+.14i)e^{-1.908it} \\ .27e^{1.914t} & .27e^{-1.914t} & -.27\, e^{1.908it} & -.27\, e^{-1.908it} \end{bmatrix}$$

3. Evaluate the first two rows ($n=\frac{1}{2}$, $r=1$) of Φ at $t=0$, the last two rows at $t=T$. (Consider two values of T, T=1.0120098 seconds and T=1.0 seconds.)

$$\Phi_c(T=1.0120098) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ .50 & -.55 & .027-.52i & .027+.52i \\ -1.05 & .020 & .14+.044i & .14-.044i \\ 1.90 & .040 & .097-.26i & .097+.26i \end{bmatrix} \triangleq \Phi_{c1}$$

$$\Phi_c(T=1.0) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ .50 & -.55 & .027-.52i & .027+.52i \\ -1.02 & .020 & .14+.040i & .14-.040i \\ 1.86 & .040 & .091-.26i & .091+.26i \end{bmatrix} \triangleq \Phi_{c2}$$

4. If $x(0)\neq 0$, $\underline{\gamma} = \Phi_c^{-1} \begin{bmatrix} x(0) \\ \underline{0} \end{bmatrix}$. If $x(0)=0$, $\underline{\gamma}$ is in terms of one element of $\underline{\gamma}$.

$$|\Phi_{c1}| = -8\times10^{-19} - 7\times10^{-9}i \qquad |\Phi_{c2}| = -2\times10^{-19} + .018i$$

So, T=1.0120078 can be used only if $x(0)=0$, T=1.0 only if $x(0)\neq 0$. (T=1.0120098 is the first conjugate print for $\nu=.0375$).

(a) For $x(0)=0$:

$$\begin{bmatrix} \text{Re } \gamma_1 \\ \text{Im } \gamma_1 \\ \text{Re } \gamma_2 \\ \text{Im } \gamma_2 \\ \text{Re } \gamma_3 \\ \text{Im } \gamma_3 \\ \text{Re } \gamma_4 \end{bmatrix}_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ .50 & 0 & -.55 & 0 & .027 & .52 & .027 \\ 0 & .50 & 0 & -.55 & -.52 & .027 & .52 \\ -1.05 & 0 & .020 & 0 & .14 & -.044 & .14 \\ 0 & -1.05 & 0 & .020 & .044 & .14 & -.044 \\ 1.90 & 0 & .040 & 0 & .097 & .26 & .097 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ -1 \\ .52 \\ -.027 \\ -.044 \\ -.14 \\ .26 \end{bmatrix} (\text{Im}\gamma_4)_1$$

$$= \begin{bmatrix} 0 & .337 & 0 & -2.337023 & 1.445 & 1 & -1.445 \end{bmatrix}^T \text{Im}\gamma_4$$

The notation is $\gamma_4=(-1.445+i)\ \text{Im}\gamma_4$, where $\text{Im}\gamma_4$ is not restricted to be real. For $\text{Im}\gamma_4=-i$:

$$\underline{\gamma}_1 \ \alpha \ \begin{bmatrix} .337 & -2.337 & 1-1.445i & 1+1.445i \end{bmatrix}^T \qquad (4.5.1)$$

(b) For $x(0)\neq 0$:

$$\underline{\gamma}_2 = \Phi_{c2}^{-1} \begin{bmatrix} x(0) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2.70 & 3.68 & -14.02 & -9.63 \\ -17.88 & -26.01 & 91.10 & 66.83 \\ 8.09-11.14i & 11.17-15.10i & -38.54+56.74i & -28.60+41.33i \\ 8.09+11.14i & 11.17+15.10i & -38.54-56.74i & -28.60-41.33i \end{bmatrix} \begin{bmatrix} x(0) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2.70 & -17.88 & 8.09 -11.14i & 8.09+ 11.14i \end{bmatrix}^T x(0) \qquad (4.5.2)$$

$$= \begin{bmatrix} .337 & -2.231 & 1.009-1.390i & 1.0093+1.390i \end{bmatrix}^T 8.01\ x(0) \qquad (4.5.3)$$

So, the solutions are:

(a) For $x(0)=0$:

$$x(t) = \mu(.337e^{1.914t} - 2.337e^{-1.914t} + (1-1.445i)e^{1.908it} + (1+1.445i)e^{-1.908it})$$
$$= \mu(.337e^{1.914t} - 2.337e^{-1.914t} + 2\cos(1.908t) + 2.891\sin(1.908t))$$

$$u(t) = -\frac{1}{2\nu}G^T\underline{\lambda} = \mu(.679e^{1.914t} + 4.238e^{-1.914t} + (2.858 + 1.764i)e^{1.908it} + (2.858-1.764i)e^{-1.908it})$$
$$= \mu(.679e^{1.914t} + 4.283e^{-1.914t} + 5.716\cos(1.908t) - 3.527\sin(1.908t))$$

where μ is chosen so that $\int_0^{1.0120098} u^2dt = 2E$

(b) For $x(0)\neq 0$:

$$x(t) = x(0)(2.701e^{1.914t} - 17.876e^{-1.914t} + (8.088-11.138i)e^{1.908it} + (8.088-11.138i)e^{-1.908it})$$
$$= x(0)(2.701e^{1.914t} - 17.876e^{-1.914t} + 16.175\cos(1.908t) + 22.276\sin(1.908t))$$

$$u(t) = x(0)(5.438e^{1.914t} + 32.42e^{-1.914t} + (22.06 + 14.32i)e^{1.908it} + (22.06-14.32i)e^{-1.908it})$$
$$= x(0)(5.44e^{1.914t} + 32.42e^{-1.914t} + 44.13\cos(1.908t) + 28.64\sin(1.908t))$$

where $\int_0^1 u^2dt$ is compared to $2E$ to see if the wrong value of ν was used.

4.6 Solutions for a Simple LSS

To examine the solutions produced by this algorithm for a LSS, start with the simplest model which contains all the characteristics of a LSS i.e. a rigid body mode and a lightly damped flex mode which may be closely spaced with respect to a second flex mode. This system is modelled as a three mass slinky as shown in Figure 4.6.1. Initially, consider a non-colocated SISO set up, with the thruster at one end and the (position) sensor at the other:

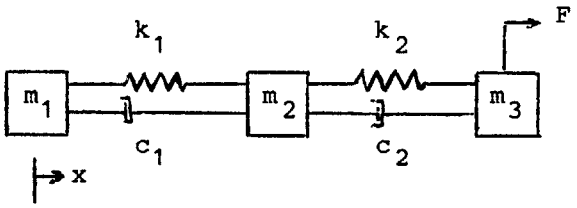


Figure 4.6.1: Three Mass, SISO Slinky

For the graphs in this section, the numerical values of the parameters are:

$m_1=m_3=100$ kg; $m_2=300$ kg ($\sum_{i=1}^{3} m_i = 500$ kg)

$k_1=k_2=100$ N/m

$c_1=c_2=2$ N-sec/m

The maximum thrust available is 50 N, which is included in the gain matrix, so that the optimal input should be <= 1. Substituting these values into the modal equations 4.3.3(4.3.1) and 4.3.4(4.3.2) gives:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\omega_1^2 & -2\zeta_1\omega_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\omega_2^2 & -2\zeta_2\omega_2 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \\ 0 \\ b_6 \end{bmatrix} \tag{4.6.1}$$

$$C = \begin{bmatrix} c_1 & 0 & c_3 & 0 & c_5 & 0 \end{bmatrix}$$

where: $b_2 = .10$ N/kg

$\omega_1 = 1.000$ rad/sec, $\zeta_1 = .01000$, $b_4 = -.15$ N/kg

$\omega_2 = 1.291$ rad/sec, $\zeta_2 = .01291$, $b_6 = .12$ N/kg (4.6.2)

$c_1 = 1$, $c_3 = 1.58$, $c_5 = 1.22$

The transformation to modal form uses 3 eigenvectors (mode shapes) which have been normalized via $\sum_{i=1}^{3} m_i \phi_j^2(i) = \sum_{i=1}^{3} m$, where $\phi_j(i) \triangleq$ ith element of the jth eigenvector. The elements of the mode shapes are the three displacements x_1, x_2, and x_3. For this system:

$$\underline{\phi}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \underline{\phi}_1 = \sqrt{\frac{5}{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \qquad \underline{\phi}_2 = \sqrt{\frac{3}{2}} \begin{bmatrix} 1 \\ -2/3 \\ 1 \end{bmatrix} \tag{4.6.3}$$

Characteristics of the Solution

Before examining optimal inputs for identification of this system, several characteristics of the solution are described. This makes the optimal inputs easier to interpret and generalize.

1. The frequency content of the optimal input does not depend on the initial condition. In addition to the elements of A, B, C, it depends on:

$\underline{x}(0) = \underline{0}$: T (or equivalently ν)

$\underline{x}(0) \neq \underline{0}$: E and T ($\nu$ can replace either E or T)

Figure 4.6.2 shows how the frequency content changes with T for an input optimized with respect to a frequency with $\underline{x}(0)=\underline{0}$.

2. As $\underline{x}(0)\to\underline{0}$, the shape of the optimal input approaches that of $\underline{x}(0)=\underline{0}$ for the same choice of T. (Compare equations 4.5.1 and 4.5.3.) The amplitudes will also match for the same choice of E.

3. For $\underline{x}(0)=\underline{0}$, large values of T require large values of ν and vice versa. As ν increases, the frequency content of the optimal input approaches the system frequencies. Hence, for large T, the optimal input is a sinusoid at the system frequency.

4. The two problems:

$$\text{Minimize: over u} \qquad J = -\frac{1}{2}\int_0^T (\underline{\xi}^T H^T K H \underline{\xi})dt$$

$$\text{Subject to:} \qquad \frac{1}{2}\int_0^T \underline{u}^T\underline{u}\, dt = E$$

and:

$$\text{Minimize: over u} \qquad J = -\frac{1}{2}\int_0^T (\underline{\xi}^T H^T K H \underline{\xi} - q\underline{u}^T\underline{u})dt$$

are mathematically exactly the same problem, where $q=2\nu$ in the notation used in previous sections. If q were less than zero, the algorithm would be trying to use a lot of input. With no boundary conditions on $\underline{\xi}$, this is like the inverse of the regulator problem i.e., drive $\underline{\xi}$ as far from the origin as possible using as much control as possible. Thus q (and ν) must be greater than zero.

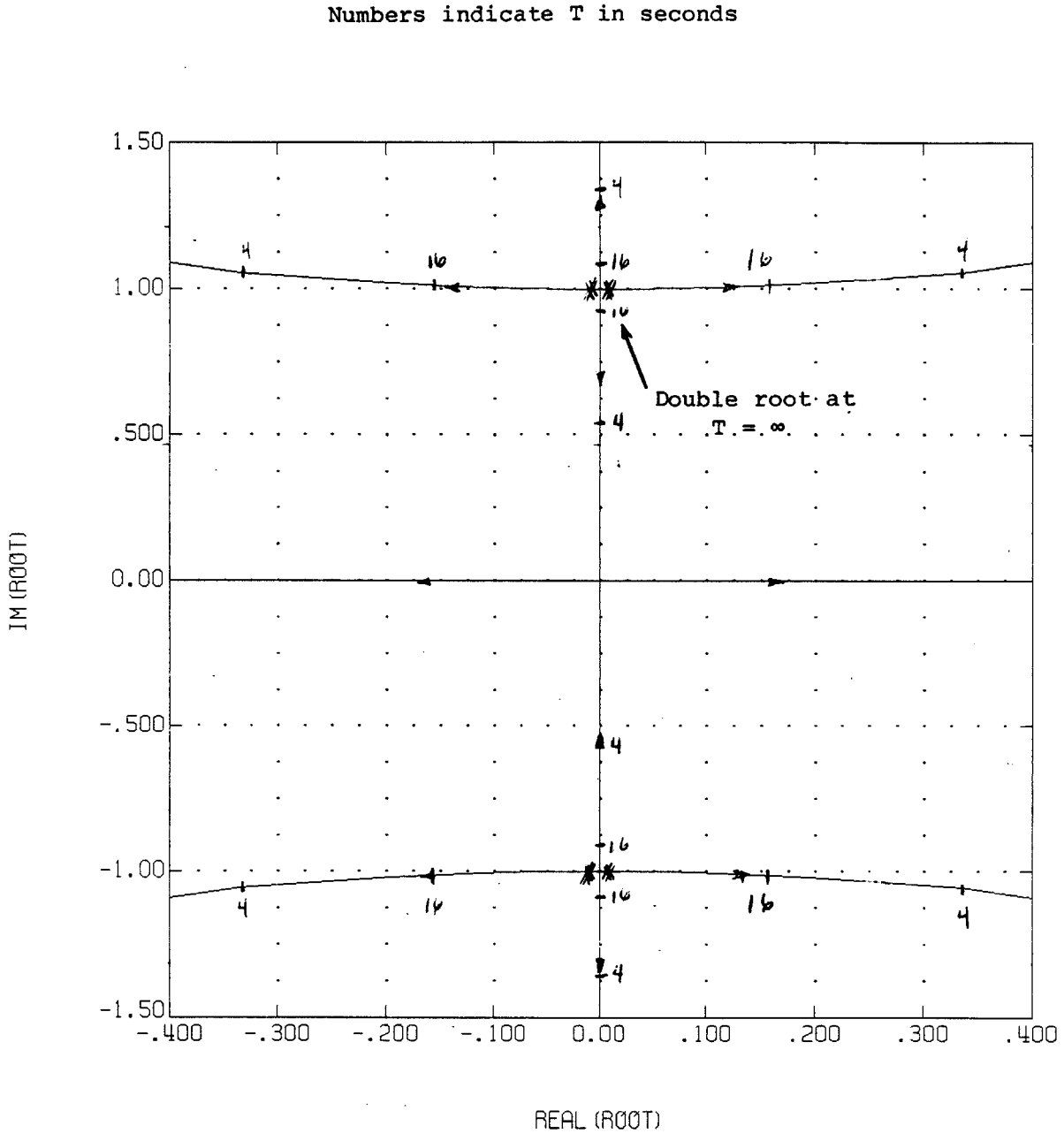


Figure 4.6.2: Frequency Content of Input Optimized With Respect to a Frequency for Different Experiment Durations

5. In the equations described by eq. 4.3.15, many of the elements of $\underline{\xi}$ and $\underline{\lambda}$ are zero for all time for a system in modal form. In addition, some elements are nonzero but not needed to calculate $\underline{u}$. Which elements these are can be identified by inspection of equation 4.3.15 or equivalently by inspection of the eigenvectors. Eliminating these elements significantly reduces the order of the problem. For example, in the most complicated case considered here (section 4.6, Optimizing with Respect to Many Parameters), the number of variables required is reduced from 96 to 38. The minimum number of variables required equals four per parameter included plus some or all of the system states.

After calculating $\underline{u}$, the deleted states can be restored if their time histories are desired.

6. For $\underline{x}(0)=\underline{0}$, the amplitude of the optimal input is determined by, and only by, the choice of E. Choosing T affects only the shape of the signal. Hence, even though the algorithm assumes E is fixed, E can be chosen so that the amplitude of the optimal input does not exceed 1. In the figures in this chapter, optimal inputs with the same T have the same E, and that E was chosen so that the input peaks near 1. Inputs with different values of T are scaled to different values of E.

Because of characteristics 2 and 6, and because in a real experiment the initial conditions will always be unknown and usually near zero, $\underline{x}(0)=\underline{0}$ was used in solving for the optimal inputs.

Optimizing with Respect to a Single Parameter

Figures 4.6.3-7 show the optimal input to maximize the sensitivity of the output to ω_1=1 rad/sec. They required assuming values for ω_1, ζ_1, b_4, and c_3 (see equation 4.6.1 and 4.6.2).

The input is, in all cases, like a damped sinusoid. For experiment durations longer than a few mode periods, the frequency of the sinusoid is nearly that of the mode. The change of the frequency content of the input with experiment duration is shown in the root locus graph in Figure 4.6.2.

Because the amount of input used is penalized, the envelope of the input decays towards the end of the experiment. For very short experiments (less than 3 mode periods), the envelope is similar to an exponential decay. The input has almost all its energy in an initial pulse, then quickly decays (see Figure 4.6.3 and 4.6.4). For longer experiments (3 to 35 mode periods), the envelope is an approximately linear decay with a more rapid decay for the last 10-15% of the experiment (see Figure 4.6.4). For longer experiment times in a damped system, the slope of the linear decay region would be so low that the output would decay rapidly if this envelope shape were maintained. Thus, as shown in Figure 4.6.5, the early part of the input drops down (enabling a steeper slope in the linear region) for longer experiment durations. Increased damping strengthens this effect, as illustrated in Figure 4.6.6. For an undamped system, this effect never appears, as shown in Figure 4.6.7. Penalizing the amplitude of the mode weakens the effect (see "Limiting State Amplitude" in section 4.6) since more rapid decay is useful in that case.

Note that the optimal signal does not always start near a peak as might be expected (see, for example, Figure 4.6.10). The trade-off between frequency content, phasing, and decay envelope usually results in starting the experiment near a peak, but not always.

Figure 4.6.8 shows the optimal input to maximize the sensitivity of the output to ζ_1=.01. The second curve on the graph shows an input optimized with respect to frequency for comparision.

frequency = 1.0 rad/sec, damping ratio = .01

Length of experiment	
———	3.14 sec
– –	6.28 "
——— ———	9.42 "

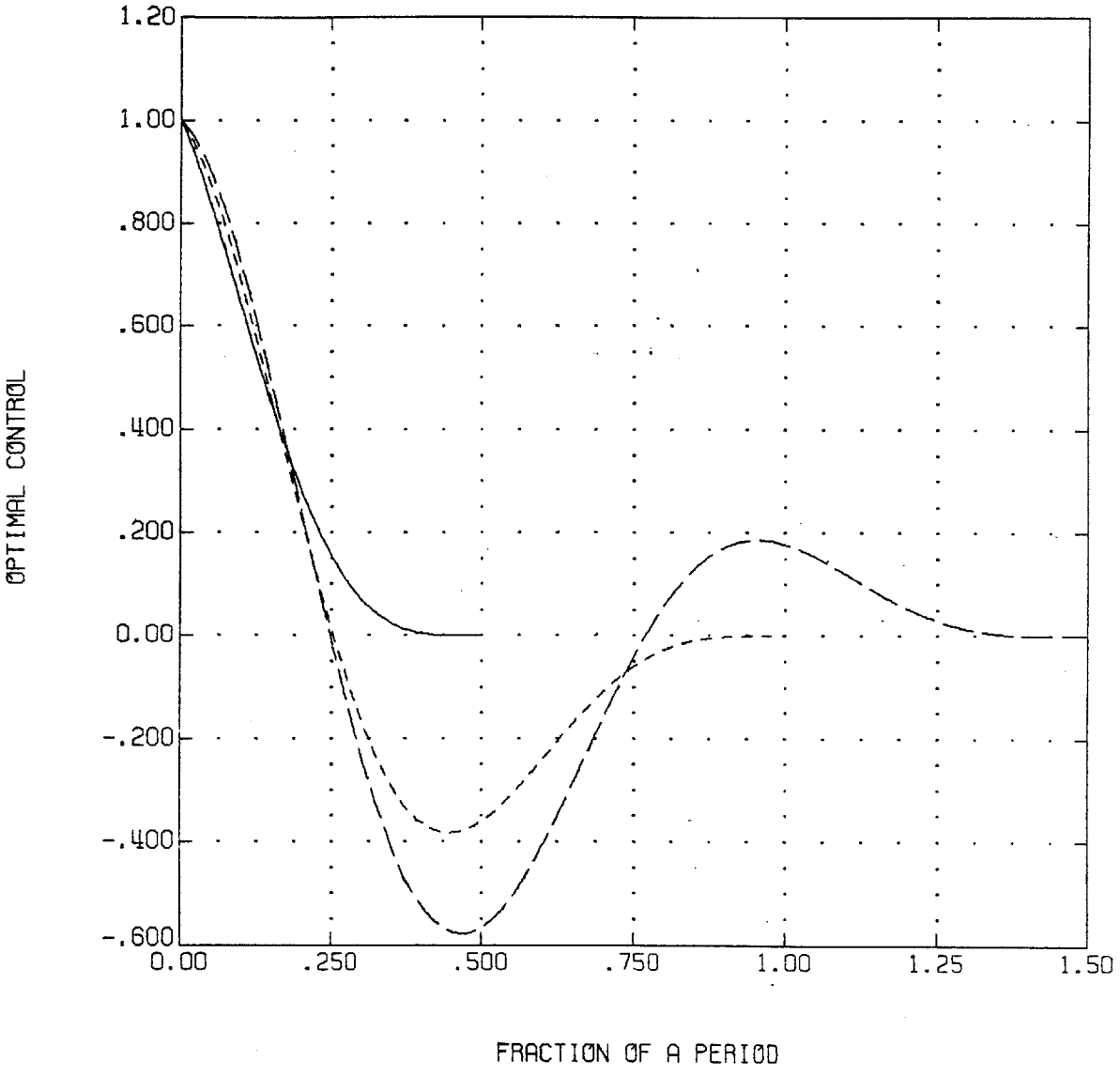


Figure 4.6.3: Input Optimized With Respect to One Frequency for Short Experiments

frequency = 1.0 rad/sec, damping ratio = .01

Length of experiment	
———	9.42 sec
– –	31.42 "
——— ———	62.83 "

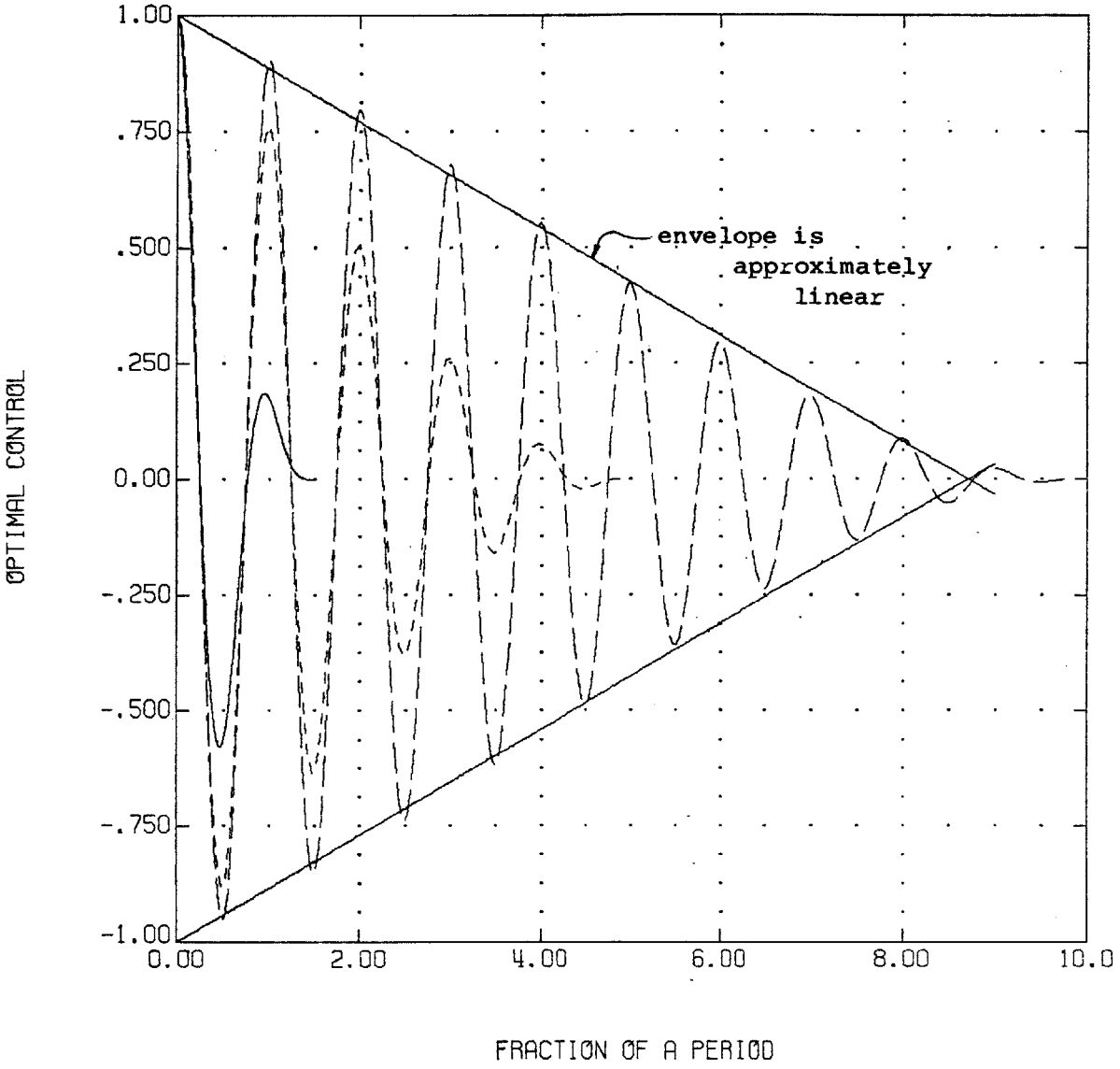


Figure 4.6.4: Input Optimized With Respect to One Frequency for Longer Experiments

frequency = 1.0 rad/sec, damping ratio = .01

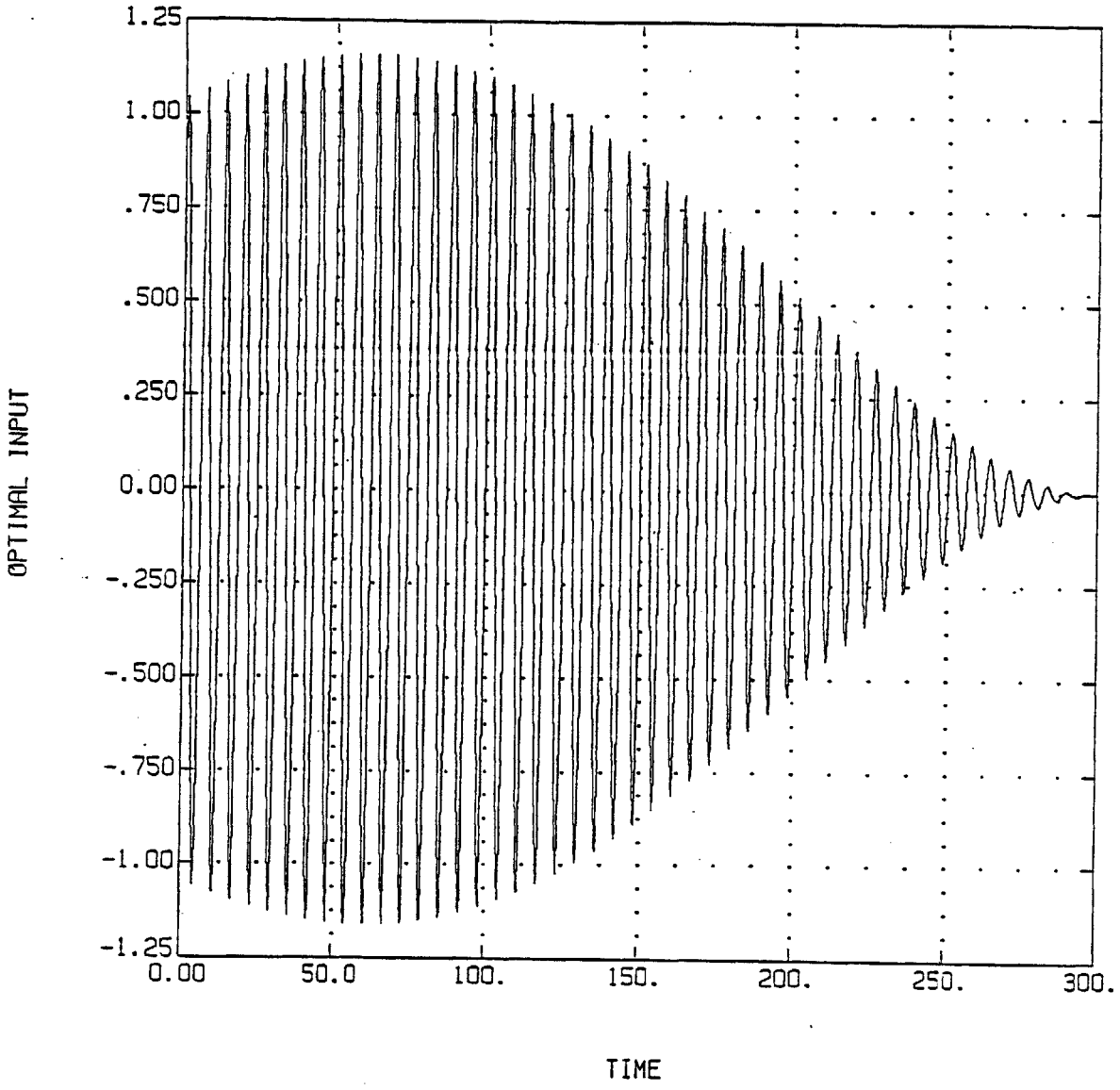


Figure 4.6.5: Input Optimized With Respect to One Frequency for a Yet Longer Experiment

frequency = 1.0 rad/sec, damping ratio = .04

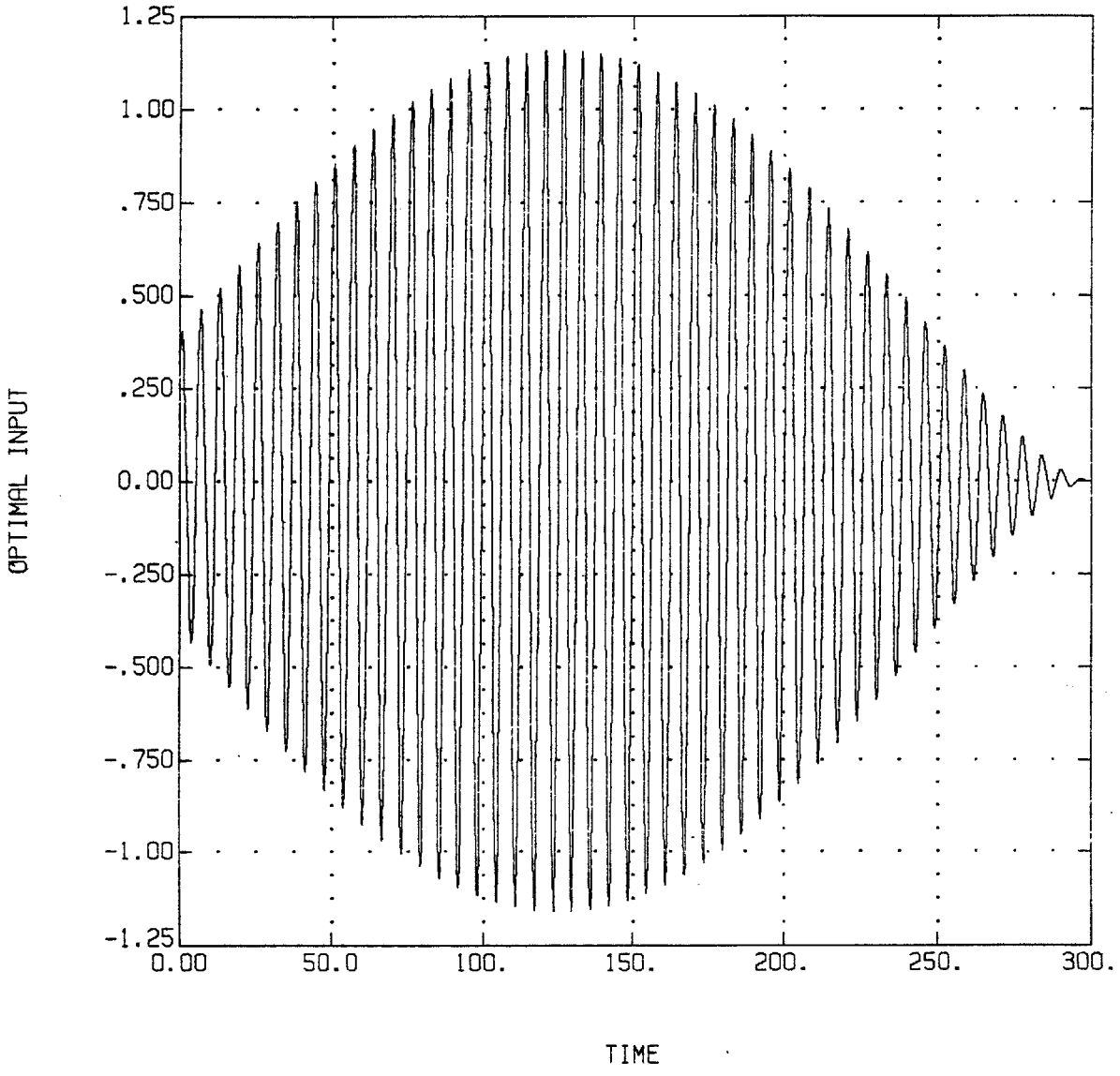


Figure 4.6.6: Effect of Increased Modal Damping on Input Optimized With Respect to a Frequency

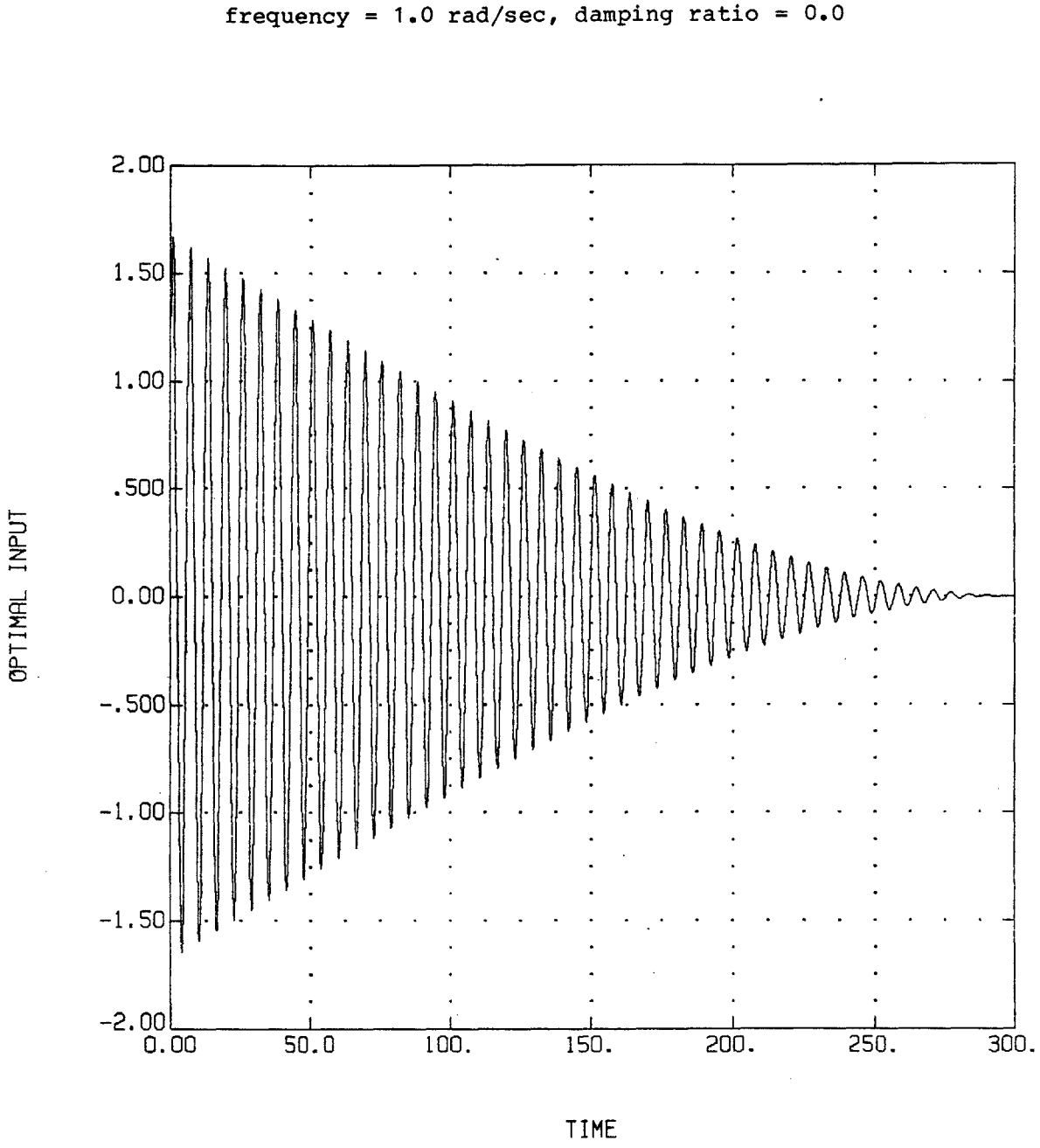


Figure 4.6.7: Effect of No Modal Damping on Input Optimized With Respect to a Frequency

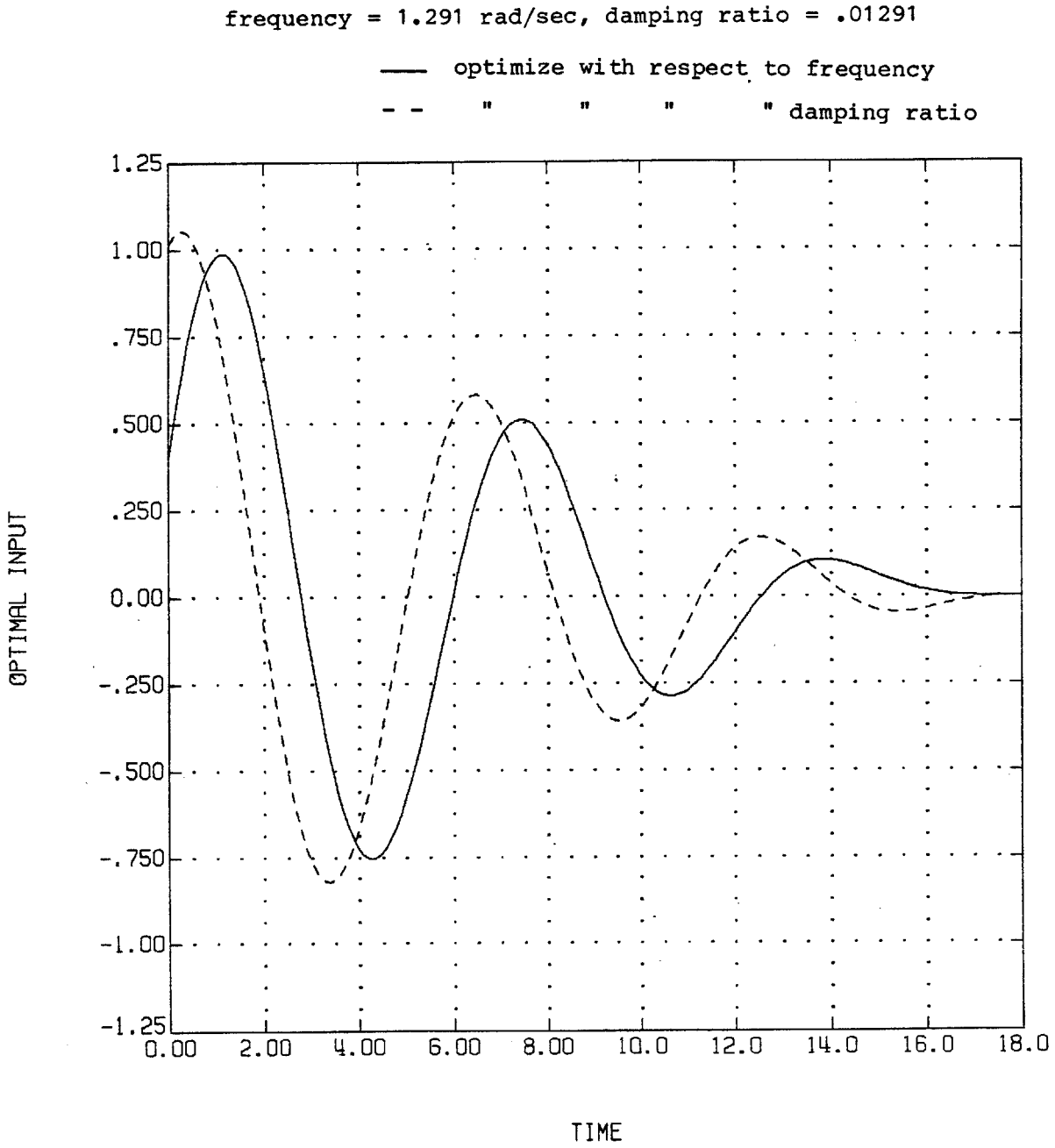


Figure 4.6.8: Input Optimized With Respect to a Damping Ratio vs. Input Optimized With Respect to a Frequency

Figure 4.6.9 shows the optimal input to maximize the sensitivity of the output to b_4=-.15 N/kg. Values for ω_1, ζ_1 and c_3 must be assumed, but not b_4.

The second curve on the graph shows an input optimized with respect to frequency for comparison. Optimizing with respect to control effectiveness shifts the input so that it is more evenly distributed over the experiment time relative to optimizing with respect to frequency.

Optimizing With Respect to Many Parameters

Figure 4.6.10 shows an example of the optimal input to maximize the weighted sensitivity of the output to ω_1, ω_2, ζ_1, ζ_2, b_4 and b_6. Values for all these parameters, and for c_1, c_3, and c_5 must be assumed.

Figures 4.6.11-13 illustrate more explicitly the effect of the weighting parameters. Figure 4.6.11 shows that for parameters in the same mode, where the single parameter shapes are similar, the optimal input is similar, too. As shown in Figures 4.6.12 and 13, when the parameters are in different modes, the resulting input is like the weighted sum of two damped sinusoids, oscillating at the two frequencies of interest.

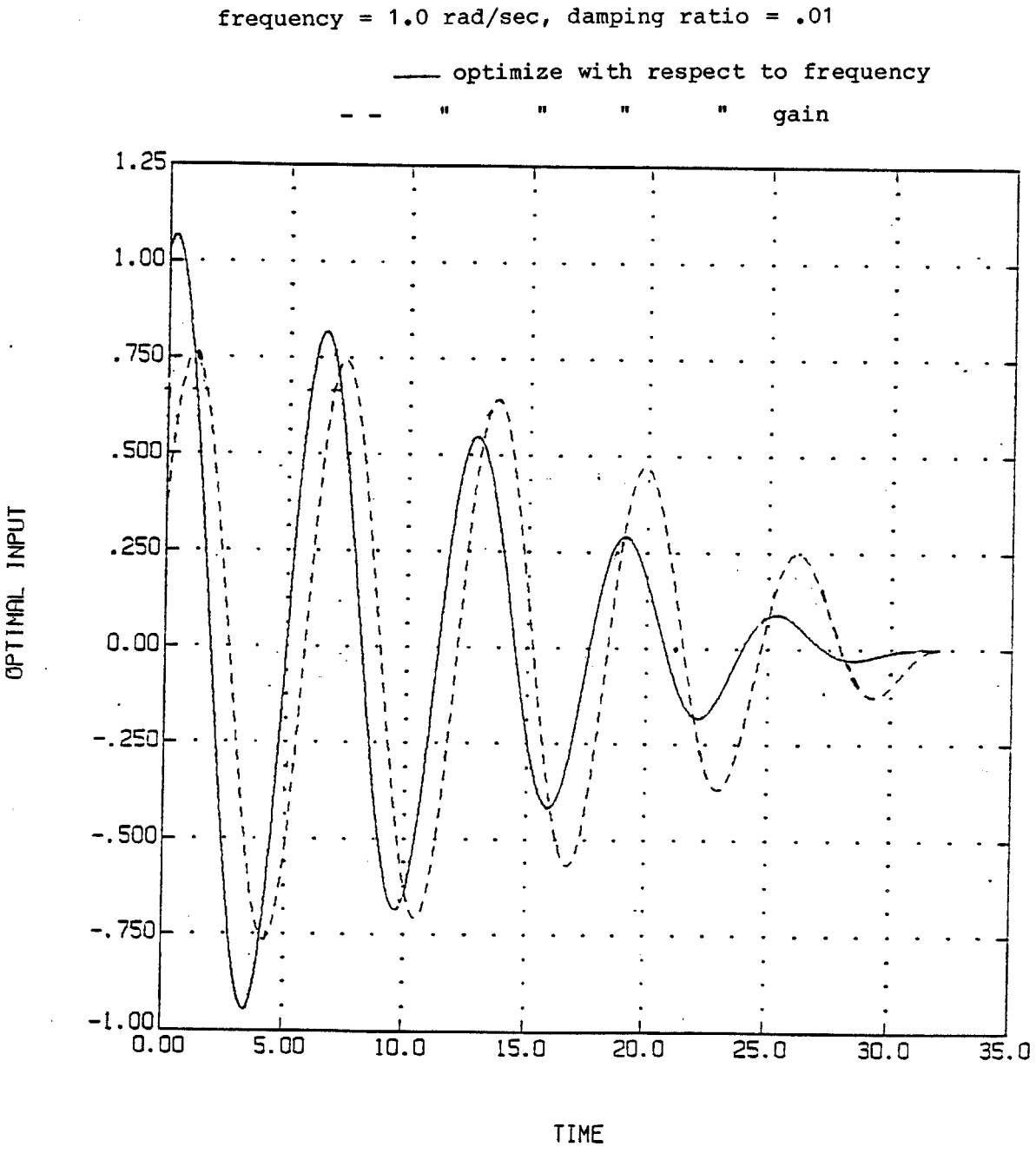


Figure 4.6.9: Input Optimized With Respect to a Gain vs. Input Optimized With Respect to a Frequency

weights: ω_1 x 1, ζ_1 x 1, b_4 x 1, ω_2 x 2,
ζ_2 x 1, b_6 x 2

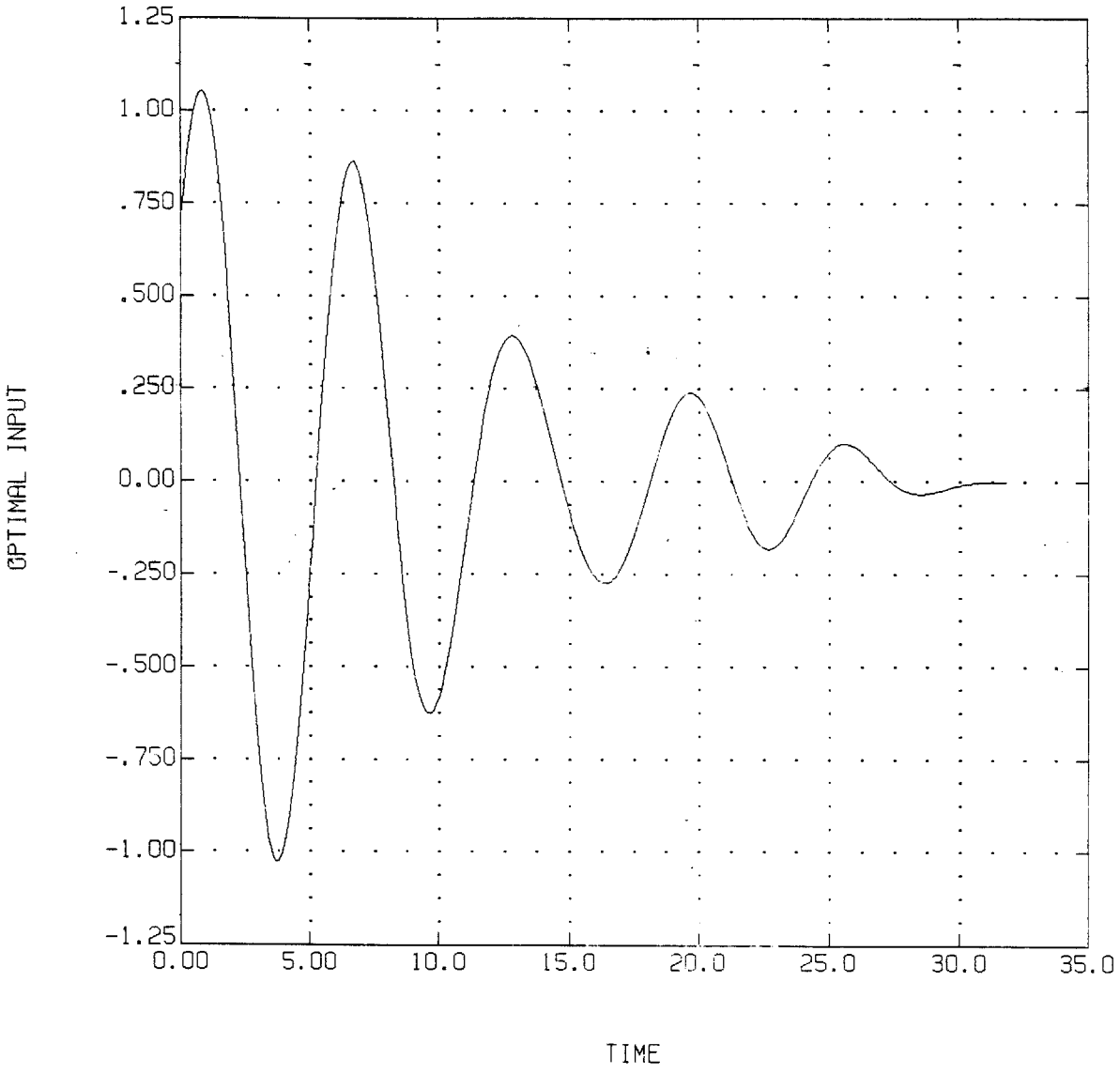


Figure 4.6.10: Input Optimized With Respect to Many Parameters

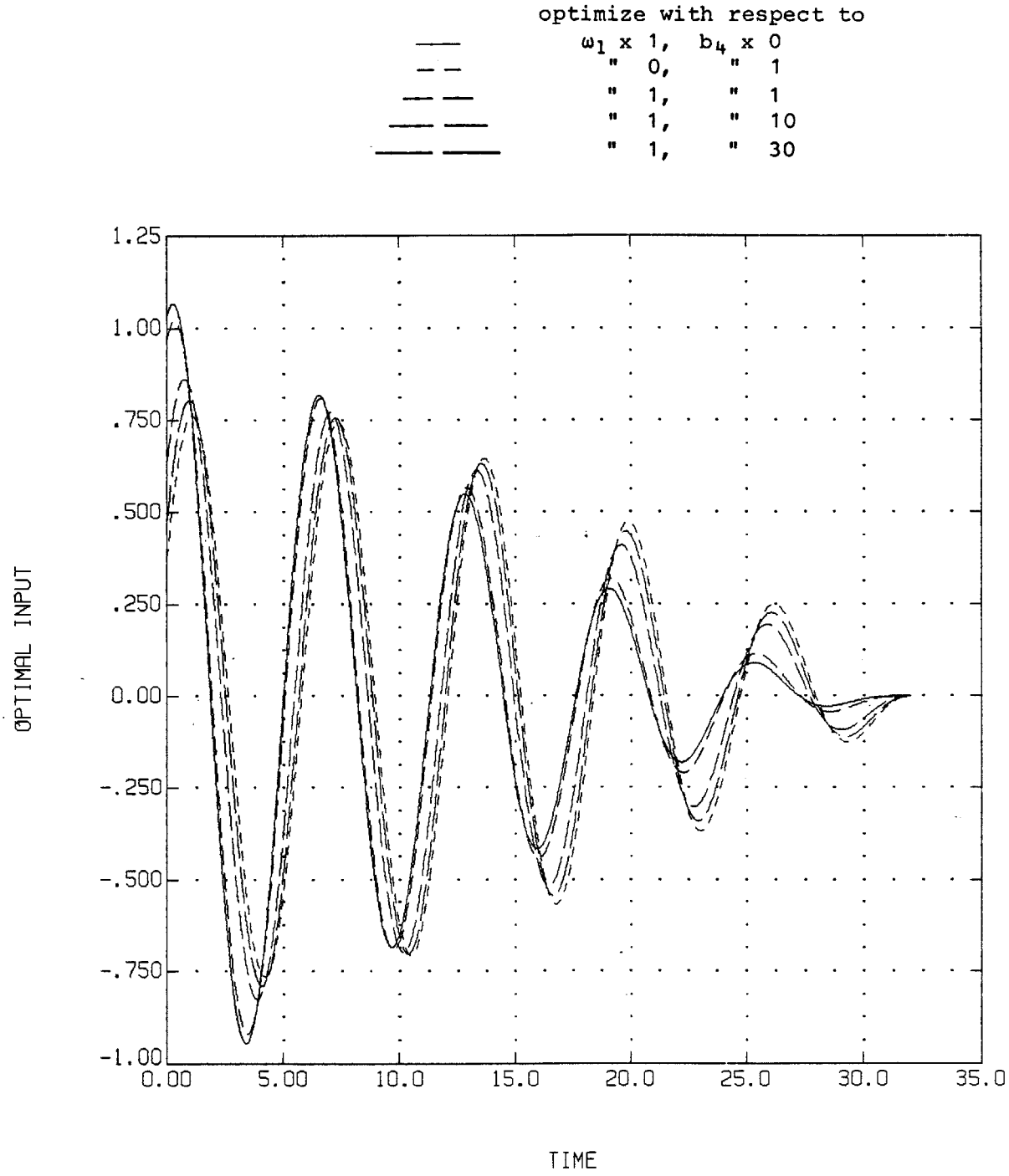


Figure 4.6.11: Effect of Changing the Relative Weighting on Two Parameters in the Same Mode

frequencies = 1.0, 4.3 rad/sec
damping ratio = .01, .01

Weights:	ω_1	ω_2
———	.006	1
– –	0	1
——— ———	1	0

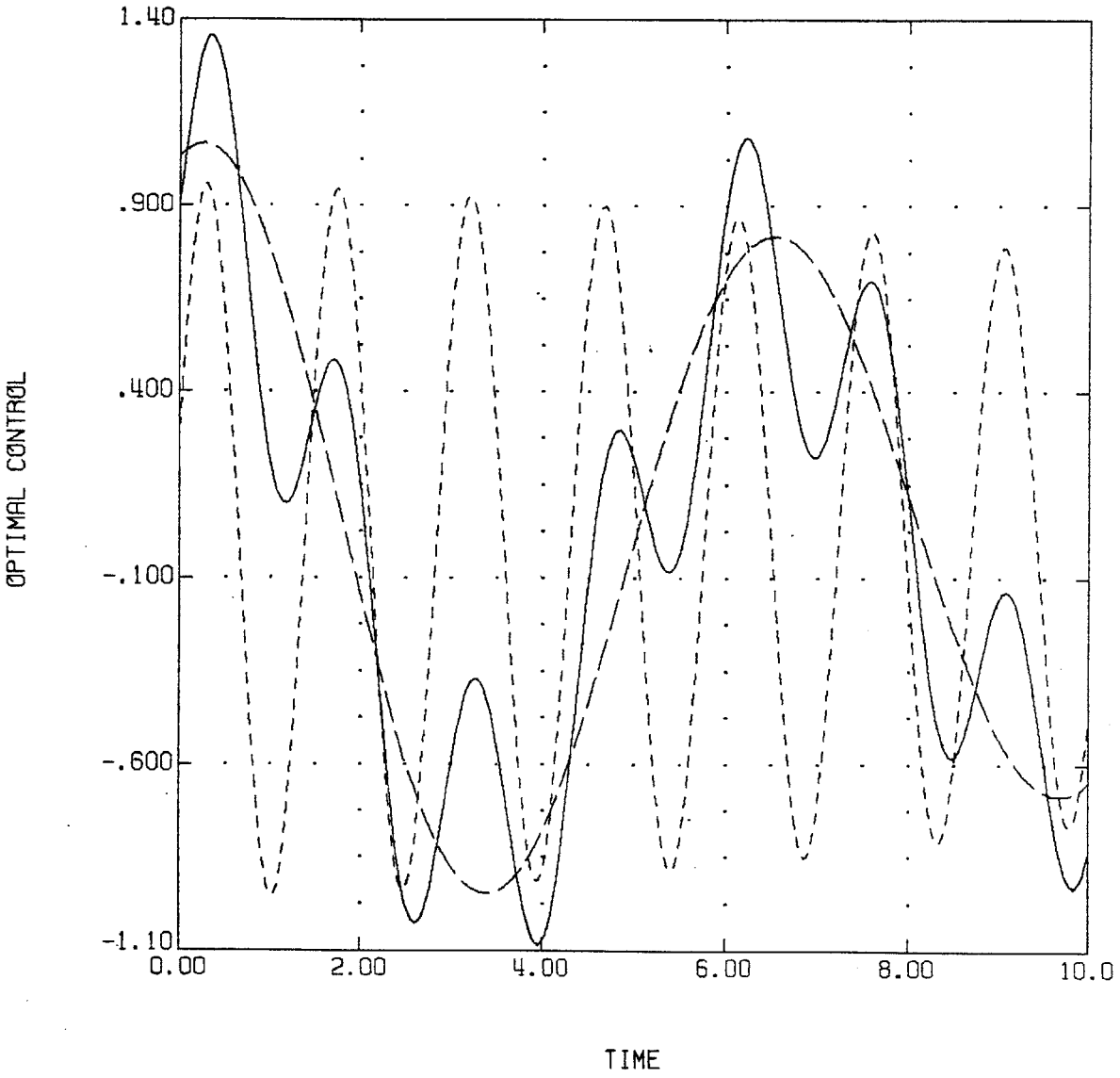


Figure 4.6.12: Input Optimized With Respect to Two Frequencies
(plot first 10 seconds of a 32 second experiment)

frequencies = 1.0, 4.3 rad/sec
damping ratio = .01, .01

	Weights: ω_1	ω_2
———	.006	1
– –	.005	1

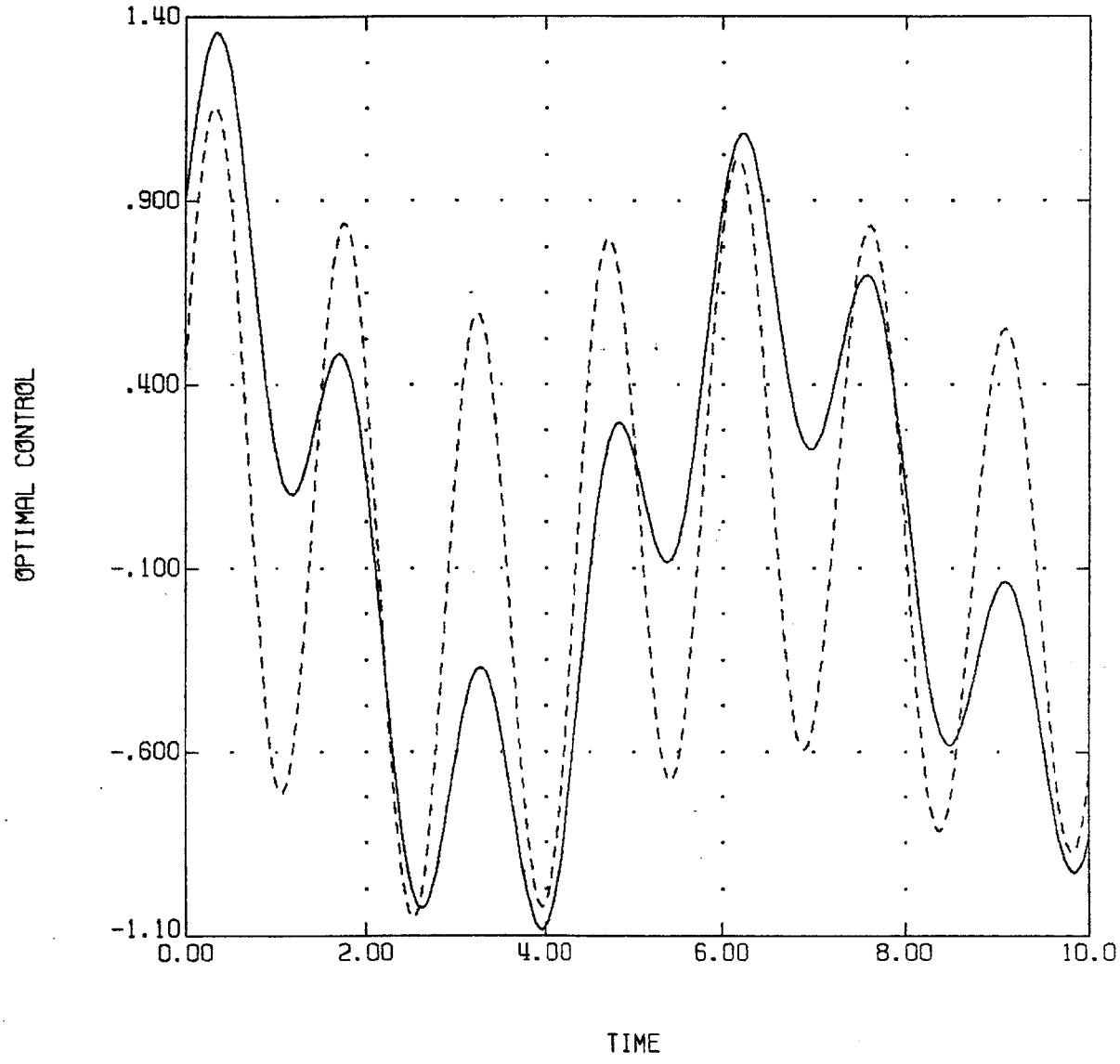


Figure 4.6.13: Effect of Changing the Relative Weighting When Optimizing With Respect to Two Frequencies (plot first 10 seconds of a 32 second experiment)

Limiting State Amplitude

One of the problems with LSSs is that they are typically lightly damped. Exciting such a structure at a natural frequency is likely to cause structural failure. To avoid this, a term is added to the cost function to penalize state amplitude i.e. equation 4.3.8 becomes:

$$J = -\frac{1}{2}\int_0^T \left(\underline{\xi}^T H^T K H \underline{\xi} - \underline{x}^T W \underline{x}\right) dt \qquad (4.6.4)$$

Since $\underline{x}$ is the first N elements of $\underline{\xi}$, the problem formulation is unchanged. Replace all H with H' and all K with K' where:

$$H' = \begin{bmatrix} I_N & 0 \\ \hline & H \end{bmatrix} \qquad K' = \begin{bmatrix} -W & 0 \\ 0 & K \end{bmatrix}$$

where the cost will be:

$$J = -\frac{1}{2}\int_0^T \left(\underline{\xi}^T H'^T K' H' \underline{\xi}\right) dt$$

This is a regulator problem where some regulation is traded off for increased sensitivity of the output to parameters. The graphs on the following pages show the results for different values of W for optimizing with respect to frequency 2 (=1.291 rad/sec).

As shown in Figure 4.6.14-16, rigid body displacement can be eliminated with very little effect on the flex modes, when optimizing over a flex mode parameter.

As shown in Figure 4.6.17-20, it is much more costly to limit the amplitude of the mode whose parameters are being identified. Figure 4.6.17 shows that the penalty shifts the linearly-decaying envelope portion of the input to earlier times, as expected (see section 4.6, Optimizing With Respect to a Single Parameter), and adds some input near the end of the run. Figure 4.6.19 shows this effect for a larger penalty. As shown in Figures 4.6.18 and 20, the penalty primarily decreases the modal response during its decay, but also decreases the peak amplitude.

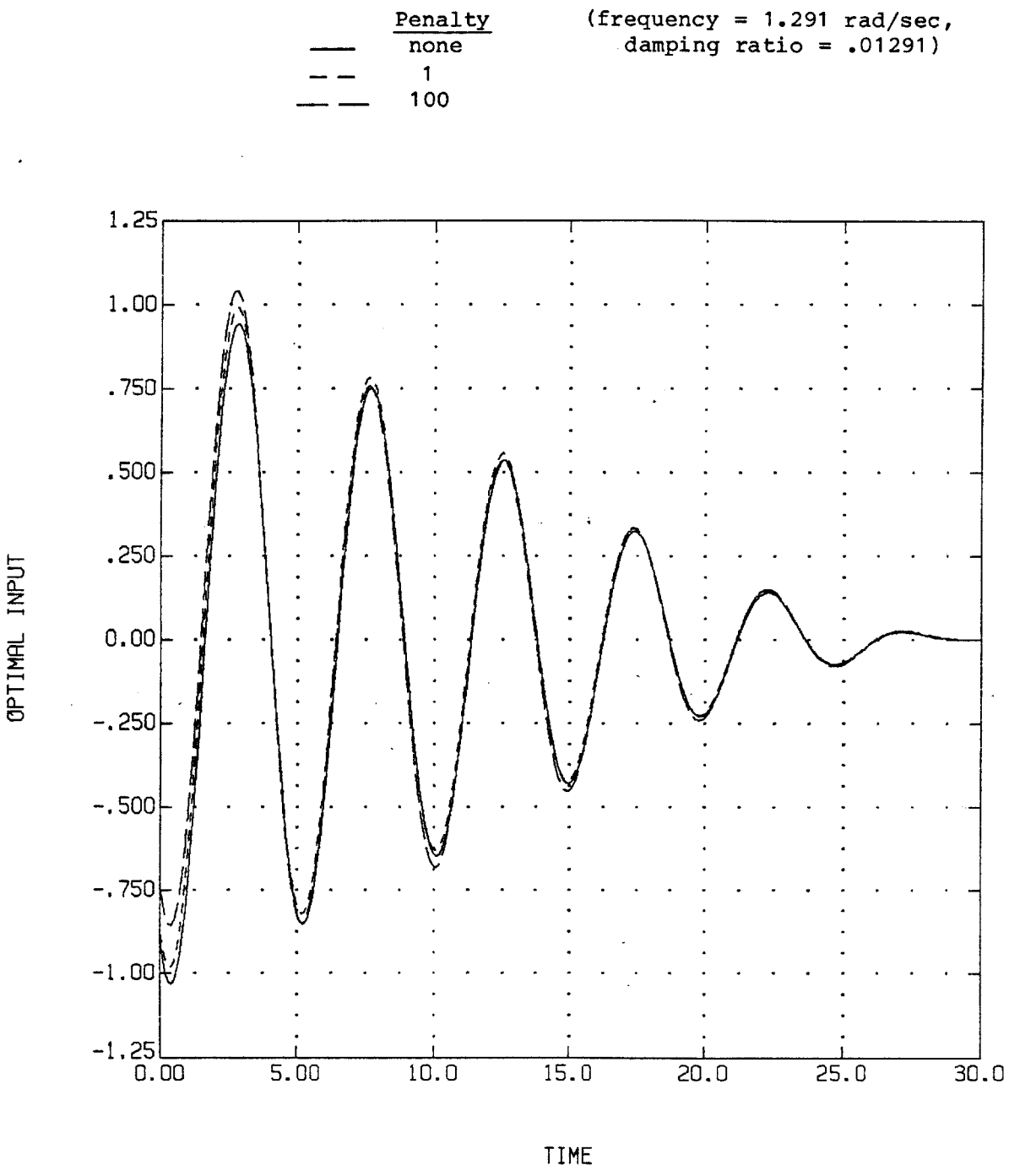


Figure 4.6.14: Input Optimized With Respect to a Frequency and Penalized for Rigid Body Motion

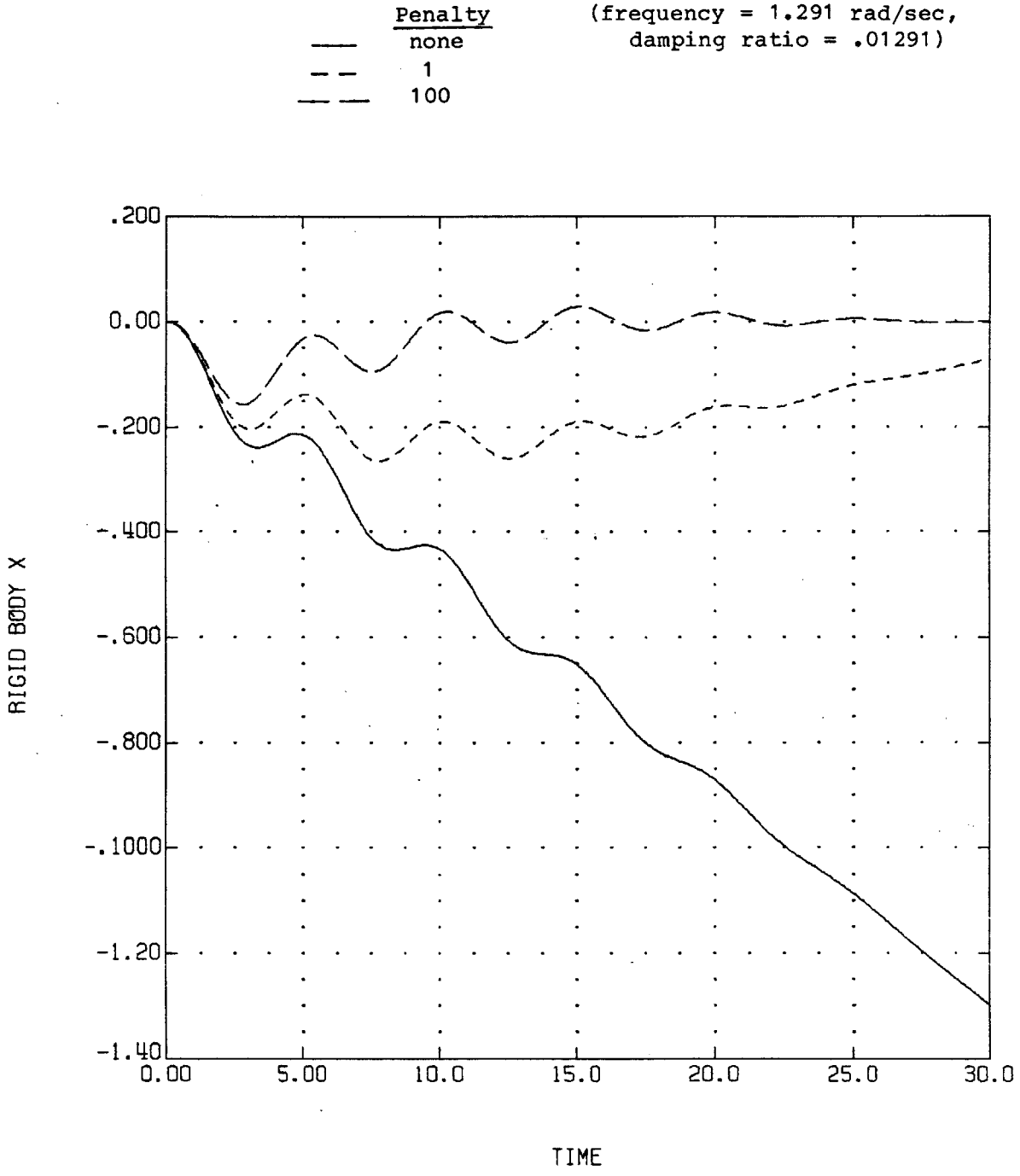


Figure 4.6.15: Rigid Body Motion with Input from Figure 4.6.14

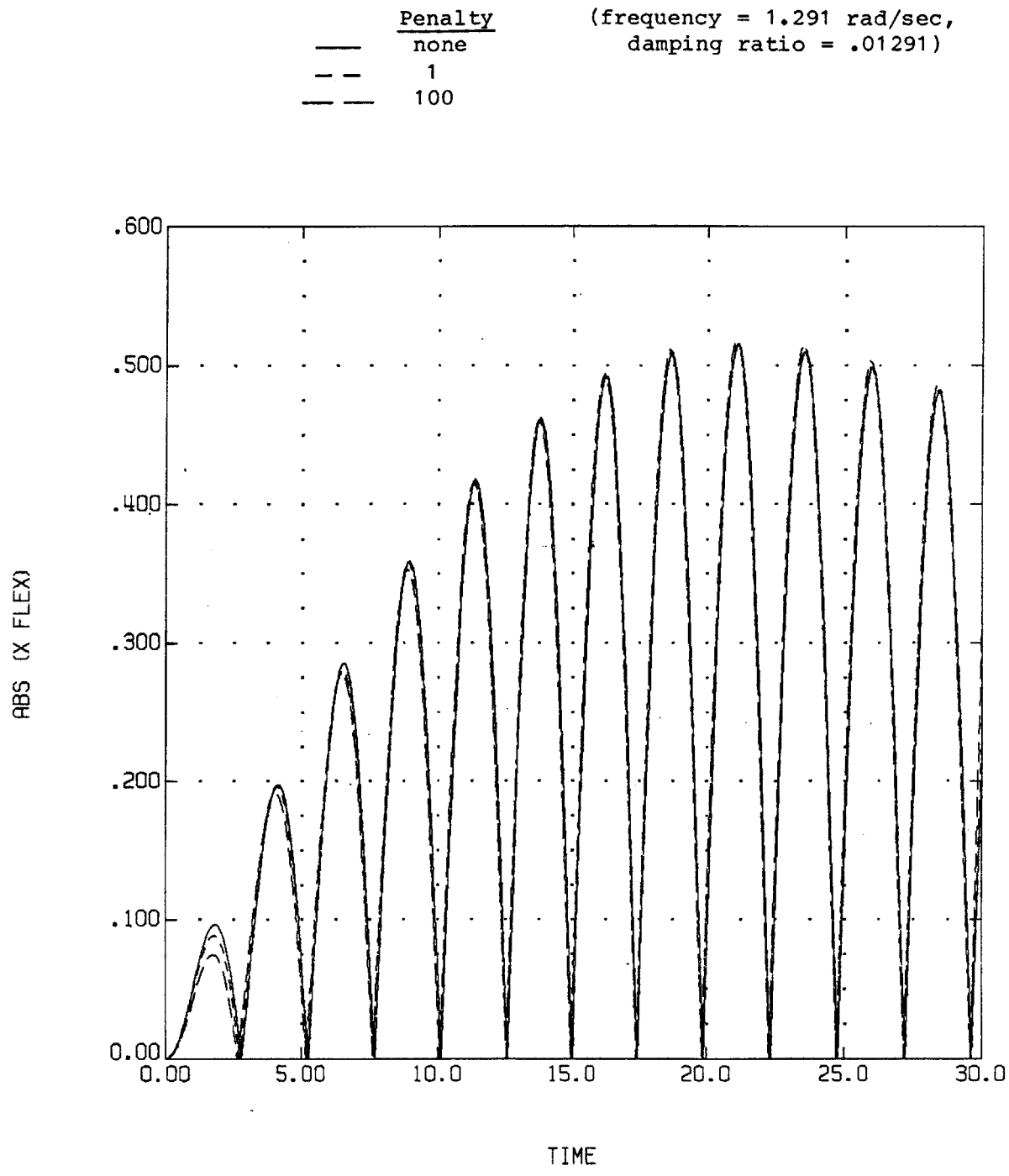


Figure 4.6.16: Flexible Mode Motion with Input from Figure 4.6.14

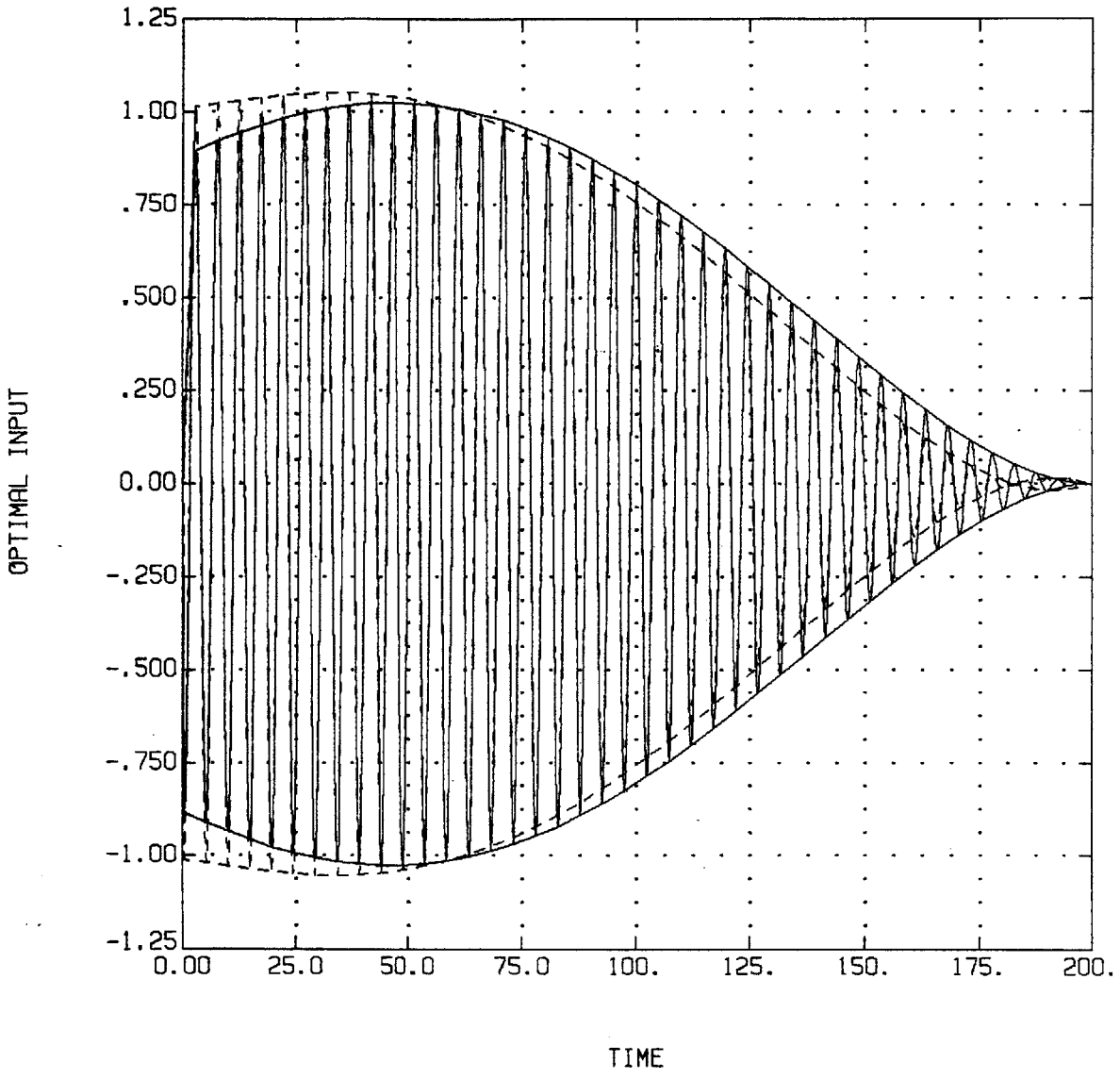


Figure 4.6.17: Input Optimized With Respect to a Frequency With a Small Penalty on a Flexible Mode

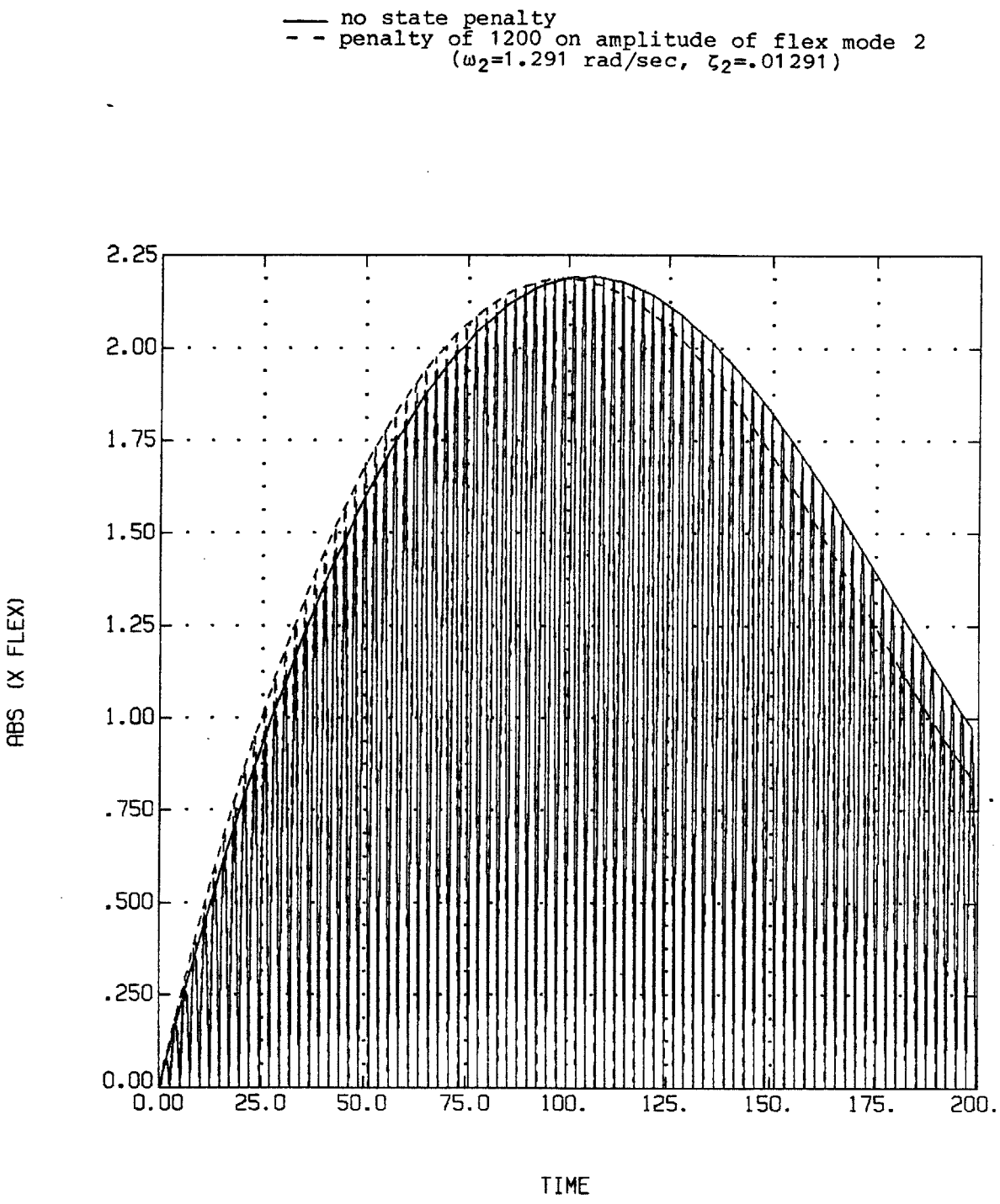


Figure 4.6.18: Modal Response to Input Optimized With a Small Penalty on a Flexible Mode

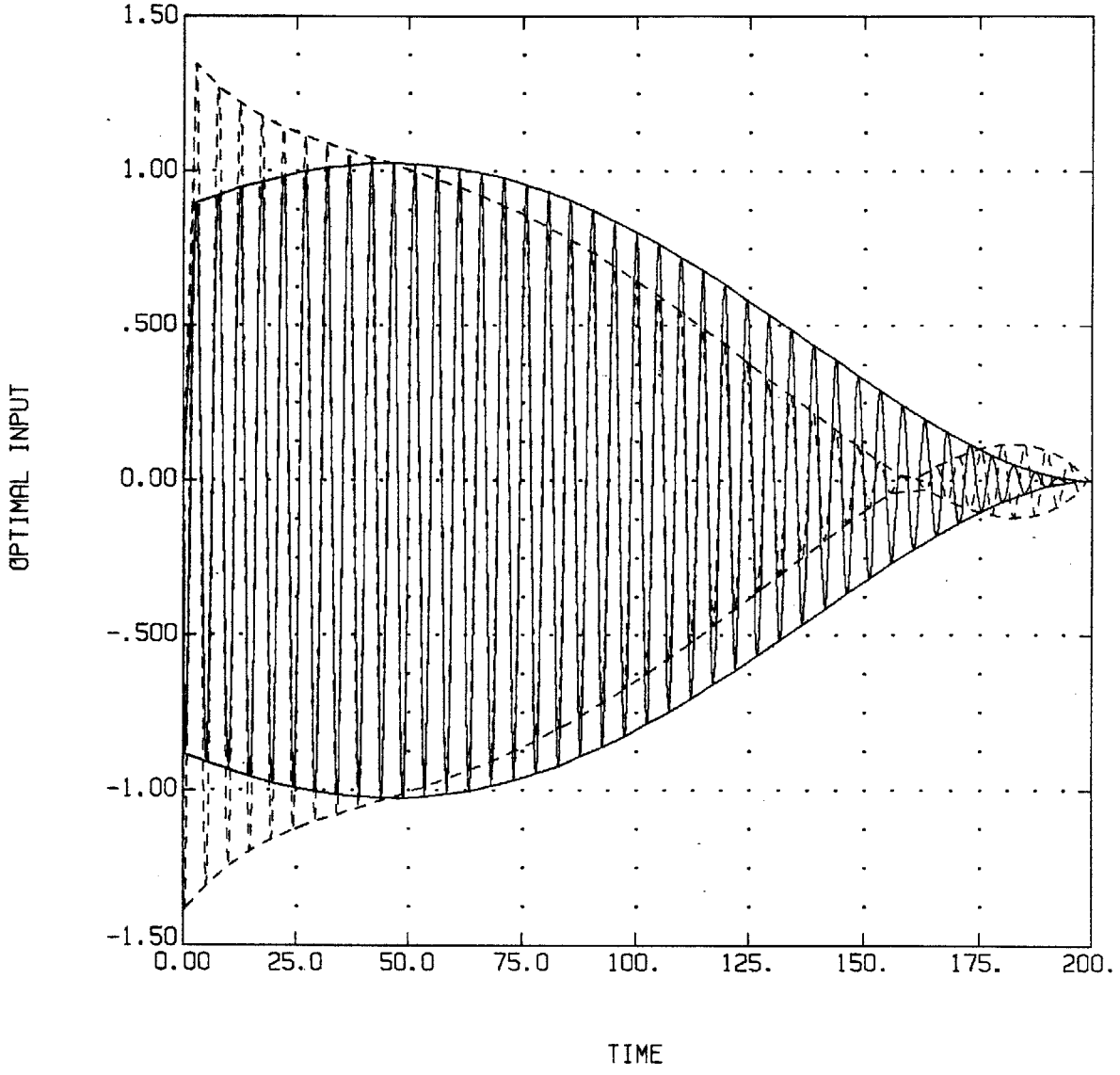


Figure 4.6.19: Input Optimized With Respect to a Frequency With a Large Penalty on a Flexible Mode

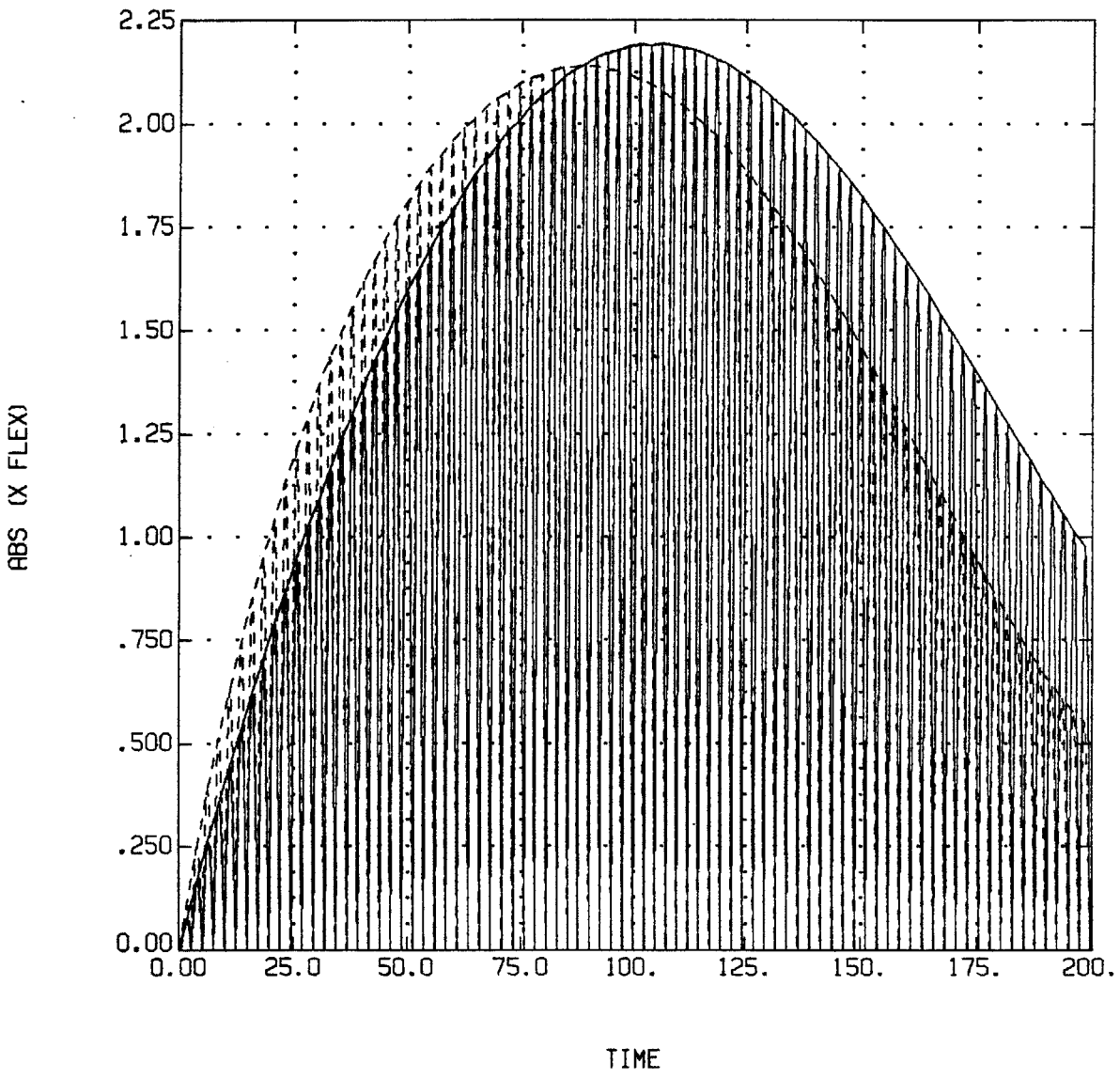


Figure 4.6.20: Modal Response to Input Optimized With a Large Penalty on a Flexible Mode

Separating Closely Spaced Frequencies

Another LSS problem is that frequencies are typically closely spaced. To highlight the frequency separation, model the frequencies as $\omega_1=\omega_c-\varepsilon$ and $\omega_2=\omega_c+\varepsilon$, then optimize with respect to ε. Note that this is the same as replacing the $\underline{y}_{\alpha_i}^T \underline{y}_{\alpha_i}$ term in the cost with $(\underline{x}_{\omega_2}^T - \underline{x}_{\omega_1}^T)\, C^T C\, (\underline{x}_{\omega_2} - \underline{x}_{\omega_1})$. Graphs on the following pages show the result. In all the graphs, the damping ratios of the two modes are .01.

In the first graph (Figure 4.6.21), the input is optimized with respect to the frequency difference of two widely spaced modes (1 and 4.3 rad/sec). The input looks much like optimizing with respect to a single frequency, because the amplitude of response of the lower frequency mode is so much higher than that of the higher frequency mode that almost all the input goes to exciting the lower mode.

Figure 4.6.22 shows the input for closer frequencies (1 and 1.1 rad/sec). As the frequencies get closer together, the input develops a beat pattern. As shown in figure 4.6.23, this beat pattern can be duplicated using the program which optimizes with respect to the frequencies rather than their difference. The solid line is optimized with respect to the two frequencies. The weighting on the two frequencies was chosen so that the integrated sensitivities of the state with respect to the two frequencies is similar for the two inputs. (Frequency 1 was weighted with 1, frequency 2 with 1.8.)

Thus, optimizing with respect to frequency difference selects the input which, among all the different inputs obtained by varying the relative weights when optimizing with respect to the two frequencies, gives the strongest beat. This is illustrated in Figure 4.6.24, which shows the measurement (with the rigid body motion subtracted out) in response to inputs optimized three different ways: (1) optimized with respect to frequency difference (solid line); (2) optimized with respect to two frequencies with both frequencies weighted by 1; (3) optimized with respect to two frequencies with frequency 1 weighted by 1 and frequency 2 weighted by 2. The strongest beat is obtained by the input optimized with respect to the frequency difference.

ω_1 = 1.0 rad/sec, ζ_1 = .01
ω_2 = 4.3 " , ζ_2 = "

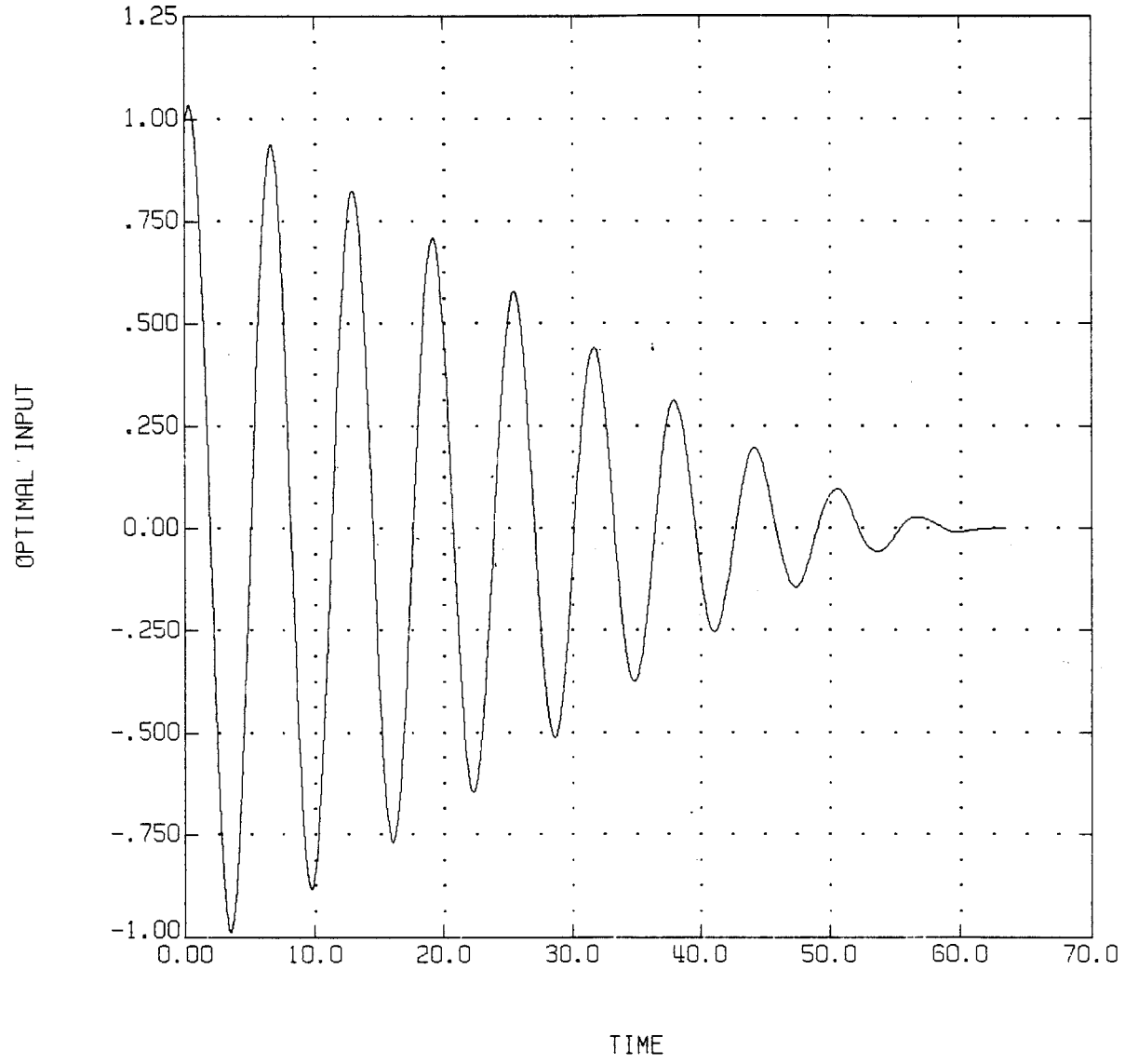


Figure 4.6.21: Input Optimized With Respect to a Frequency Difference for Widely Spaced Modes

ω_1 = 1.0 rad/sec, ζ_1 = .01
ω_2 = 1.1 " , ζ_2 = "

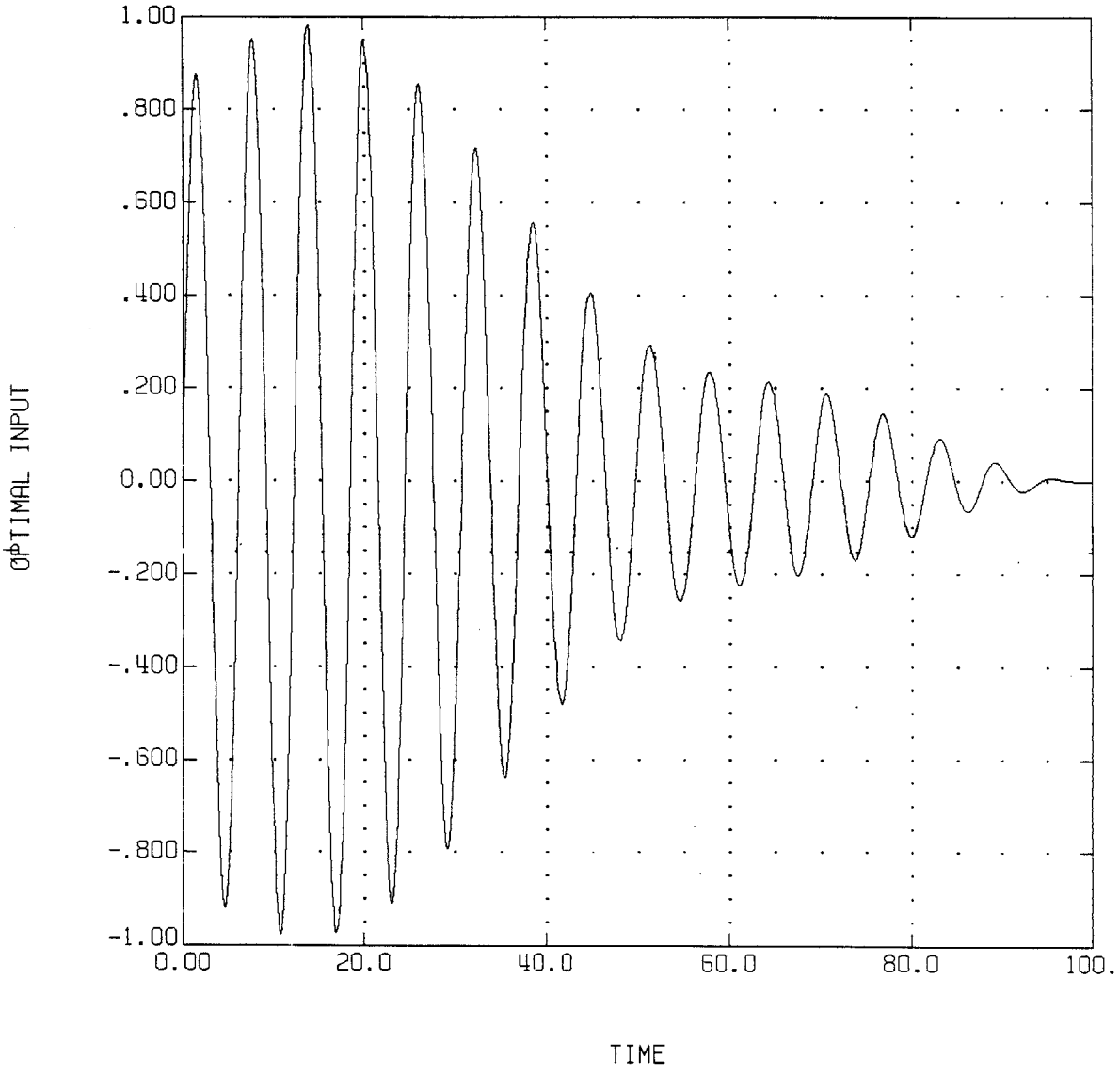


Figure 4.6.22: Input Optimized With Respect to a Frequency Difference for Closely Spaced Modes

ω_1 = 1.0 rad/sec, ζ_1 = .01
ω_2 = 1.1 " , ζ_2 = "

—— optimize with respect to ω, penalties = ω_1 x 1, ω_2 x 1.8
- - optimize with respect to frequency difference

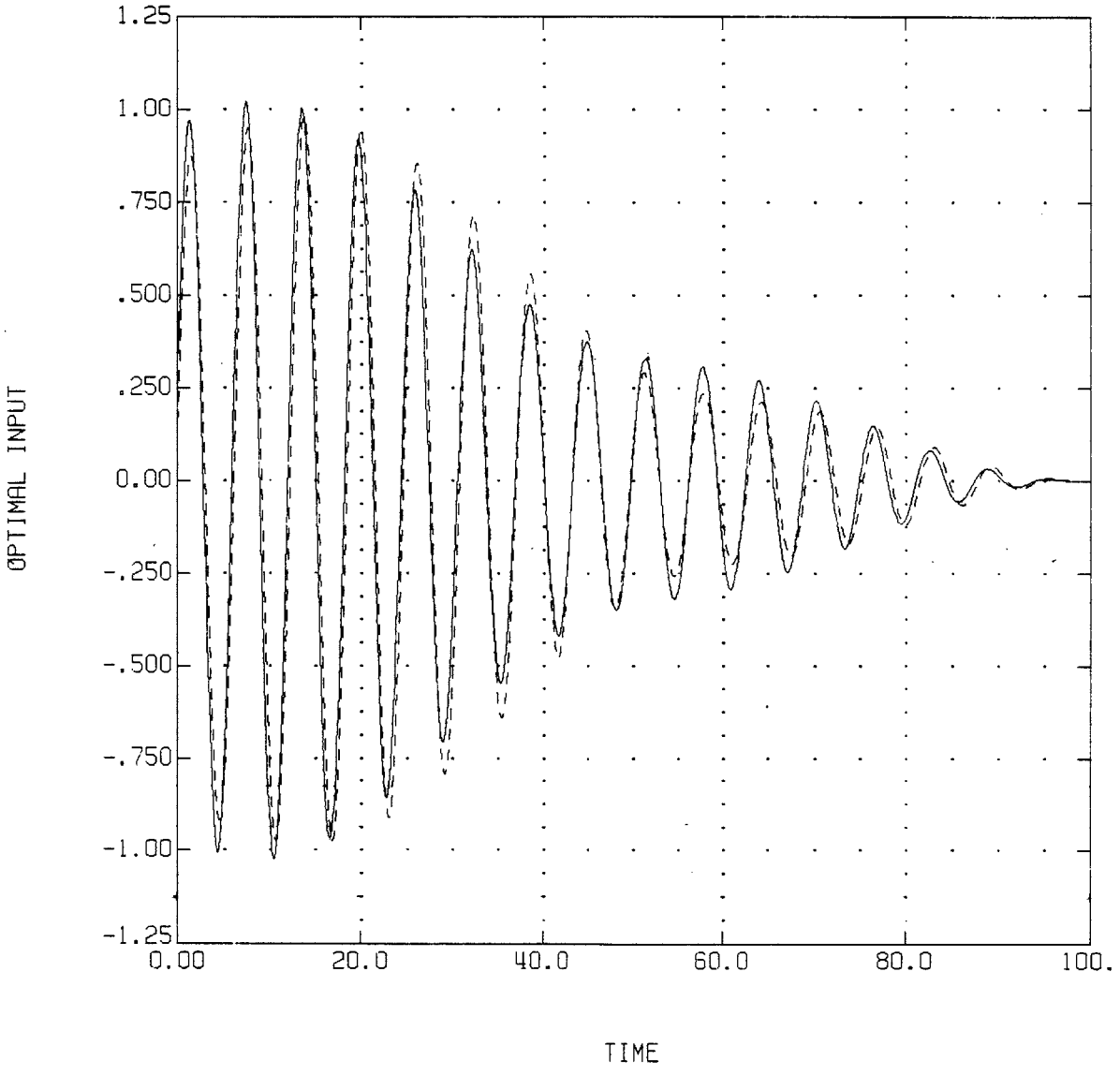


Figure 4.6.23: Duplicating Results of Optimizing With Respect to Frequency Difference By Choice of Weighting Parameters

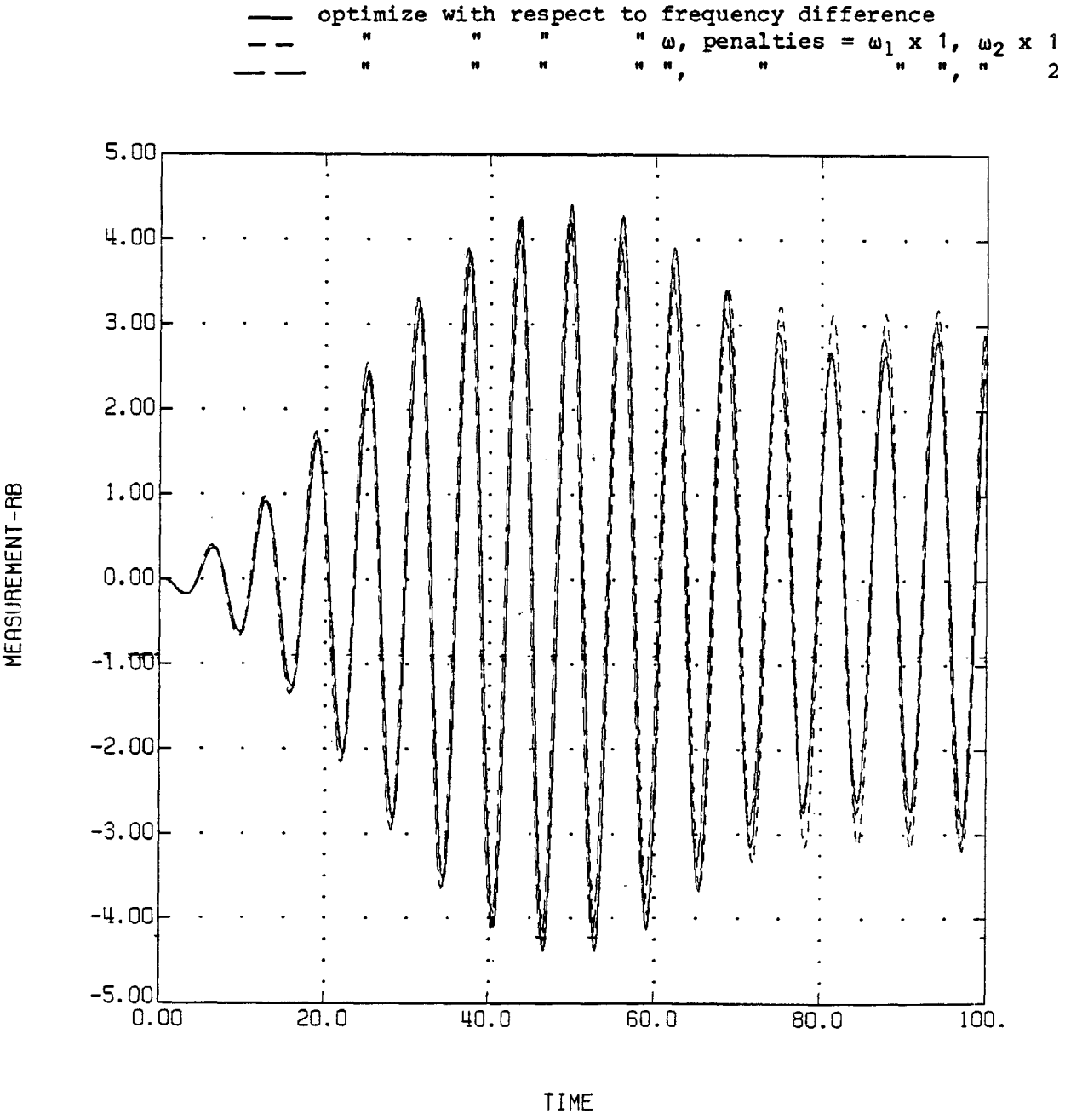


Figure 4.6.24: Output in Response to Input Optimized With Respect to Frequency Difference

4.7 Comparison with PRBS

One of the limitations of picking an input based on maximizing a cost function is that expected values of the system parameters must be assumed to calculate the optimal input. This is characteristic of all optimal input algorithms. The alternative is to use an input which requires no knowledge of the system parameters. It will be less effective but robust with respect to differences between the actual and modelled systems. A PRBS is the most common example of this type of input. The input is always at full positive or negative thrust, and switches at random at fixed time intervals.

How far can the real parameter values deviate from their expected values before the optimal input is no better than a PRBS? This was tested by simulating the system response to the optimal input and to a PRBS for different values of the system parameters.

A typical result is shown on the next page. Three optimal inputs were tested: (1) optimized with respect to frequency (at 1 rad/sec) (solid line); (2) optimized with respect to two closely spaced frequencies (at 1 and 1.291 rad/sec) (longest dashes); and (3) optimized with respect to two widely spaced frequencies (at 1 and 4.3 rad/sec) (second longest dashes). The PRBSs tested may switch every second. The gray region indicates the range of system response for all PRBSs which are restricted to switch at least every 4 seconds. (The restriction raises the lower edge of the region but has no effect on the "best" sequences.) In addition, the response to a particular PRBS is graphed (second shortest dashes). This PRBS gave the best result, of those considered, at the nominal frequency of 1 rad/sec.

As shown in Figure 4.7.2, choosing a "good" PRBS is not trivial even for this simple case. The graphs show three PRBSs with a sinusoid at the actual frequency superimposed to show how closely they match. The top graph looks fine but has an integrated weighted sensitivity of 2056, 10% poorer than the best. The middle graph is the natural guess for best, and has a sensitivity of 2222, 7% poorer than the best. The bottom graph is the best, with a sensitivity of 2398. When the system has several actuators and widely spaced modes, the only way to find a good PRBS is trial and error.

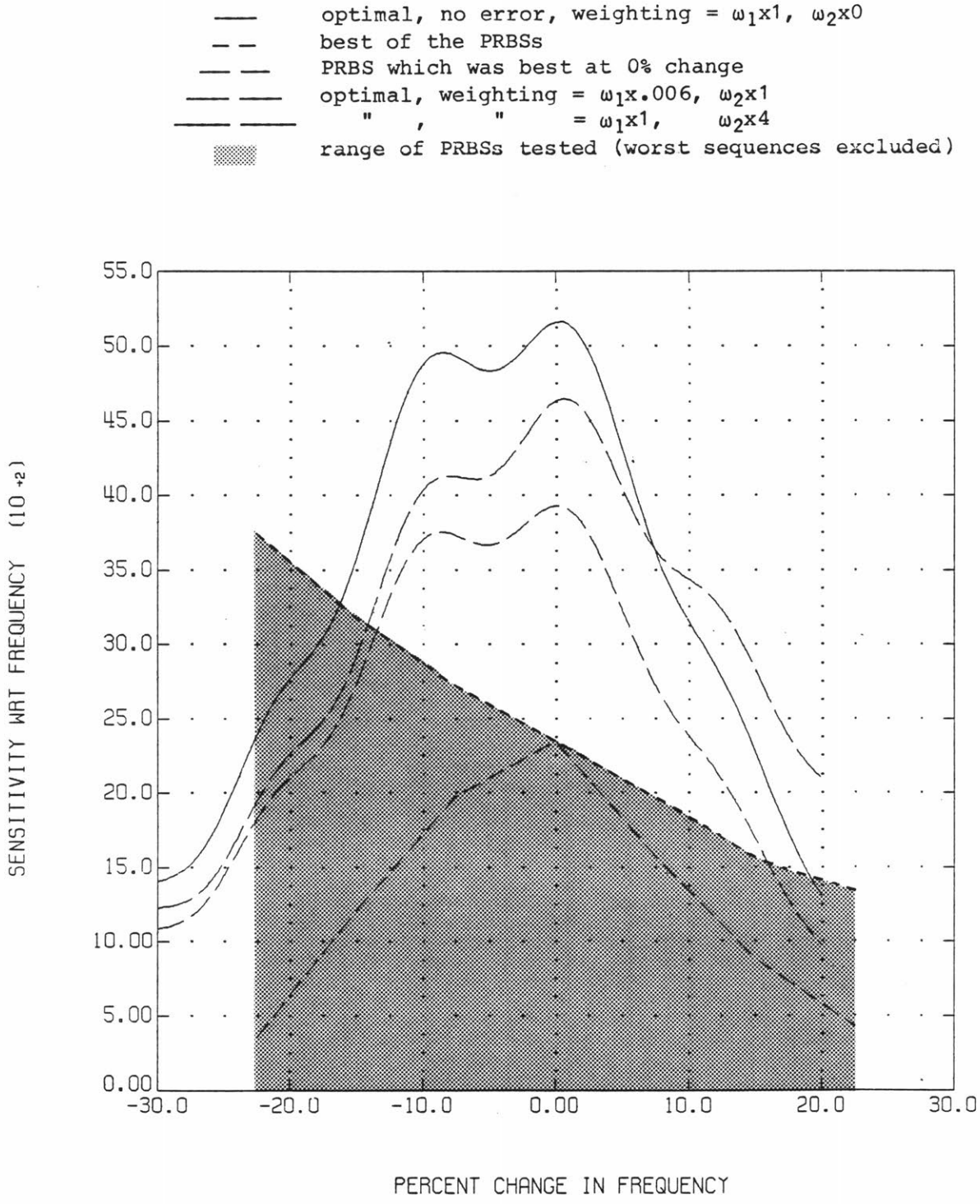


Figure 4.7.1 Comparison of Optimal Input To PRBS

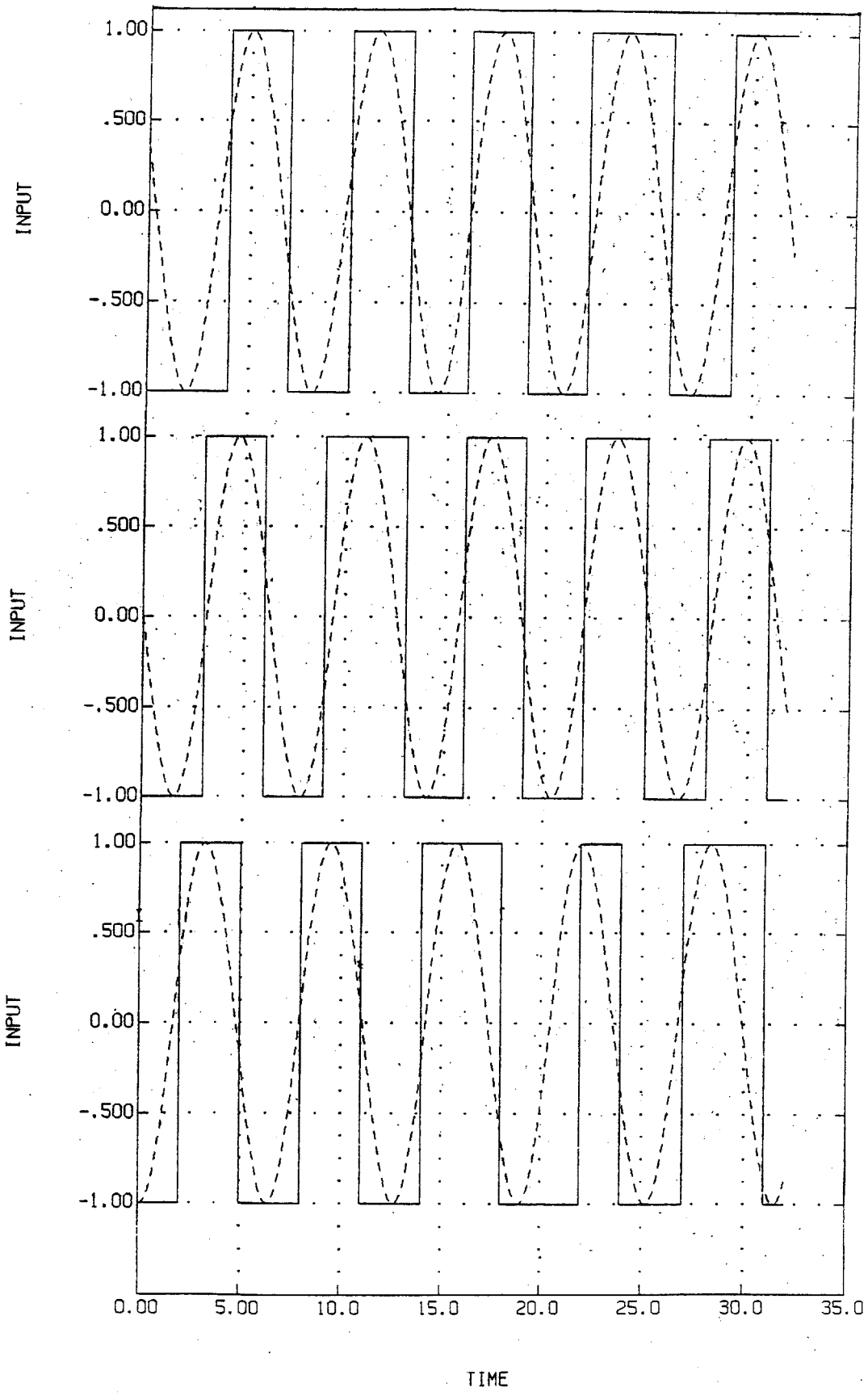


Figure 4.7.2 Comparison of Three PRBSs

How does the difference in integrated sensitivity shown in Figure 4.7.1 translate into identifier accuracies? This is demonstrated in Figure 4.7.3. The optimal input and 3 different PRBSs were used to excite the 8 mass slinky. The PRBSs were chosen at random; all of the 'best guess' PRBSs tried (such as using 1 when the optimal input would have been positive and -1 when it would been negative) were any better. These inputs are shown in Figures 4.7.4-6. To give the same integrated input (equation 4.2.3), the PRBSs have an amplitude equal to .37 that of the optimal input. The optimal identifier used the resulting input/output data to identify a model.

Figure 4.7.3(a) compares the frequency estimates for the four sets of inputs when the same noise variance is used for all four and there are no deadbands imposed to prevent breakage or drift. The results are that the PRBSs are slightly better.

However, this is not a good comparison because the identifier performance is very sensitive to SNR. The PRBSs have a much higher SNR (1000, 90 if the rigid body modes are not included in the SNR vs. 200 for the optimal input, 35 if the rigid body modes are not included). In Figure 4.7.3(b), the noise variance was varied for the PRBS runs so that the flex-only SNR was comparable. The relative performance in (b) agrees well with that expected from Figure 4.7.1.

In reality, the noise variance will not usually vary with the input. However, there will be deadbands which will prevent the PRBSs from exciting the modes to such high amplitudes. This case is examined in Figure 4.7.3(c). Again, the relative performance agrees with Figure 4.7.1.

To see how the performance degrades in the presence of modelling errors, these four sets of inputs were used to identify an 8 mass slinky with masses 55, 45, 55, 45, 55, 45, 55, 55 (vs. all 50), springs 55, 55, 55, 55, 45, 45, 55 (vs. all 50), and dampers 1.1, 1.1, .9, .9, .9, 1.1, 1.1 (vs. all 1). The resulting frequency estimates are shown in Figure 4.7.7. Comparison with Figure 4.7.3(c) shows that the optimal input estimates are degraded (as are the PRBSs, although they could go either way).

Mode	True Damped Frequency	Optimal Estimate	% Error	3 PRBS Estimates	% Error
1	0.0000	0.0024		0.0008,0.0011,0.0005	
2	0.3902	0.3835	1.8	0.3901,0.3903,0.3905	0.0,0.0,0.0
3	0.7654	0.7679	0.3	0.7655,0.7653,0.7653	0.0,0.0,0.0
4	1.1111	1.1165	0.5	1.1155,1.1142,1.1149	0.4,0.3,0.3
5	1.4142	1.4263	0.8	1.4354,1.4282,1.4290	1.5,1.0,1.0
6	1.6693	1.6693	0.4	1.6671,1.6649,1.6716	0.3,0.1,0.1
7	1.8478	1.8503	0.1	1.8692,1.8652,1.8697	1.2,0.9,1.2
8	1.9616	1.9688	0.4	1.9655,1.9630,1.9616	0.0,0.1,0.0

(a) Same integrated input, same noise variance, no deadbands

Mode	True Damped Frequency	Optimal Estimate	% Error	3 PRBS Estimates	% Error
1	0.0000	0.0024		0.0017,0.0013,0.0008	
2	0.3902	0.3835	1.8	0.3902,0.3908,0.3914	0.0,0.2,0.3
3	0.7654	0.7679	0.3	0.7659,0.7659,0.7654	0.3,0.3,0.3
4	1.1111	1.1165	0.5	1.1241,1.1232,1.1260	1.2,1.1,1.3
5	1.4142	1.4263	0.8	1.4690,1.4744,1.4756	3.9,4.2,4.3
6	1.6629	1.6693	0.4	1.6750,1.6765,1.6992	0.7,0.8,2.2
7	1.8478	1.8503	0.1	1.9064,1.9254,1.9310	3.2,4.2,4.5
8	1.9616	1.9688	0.4	1.9718,1.9662,1.9667	0.5,0.2,0.2

(b) Same integrated input, same flex SNR, no deadbands

Mode	True Damped Frequency	Optimal Estimate	% Error	3 PRBS Estimates	% Error
1	0.0000	0.0088		0.0018,0.0027,0.0094	
2	0.3902	0.3912	0.3	0.3902,0.3903,0.3907	0.0,0.0,0.1
3	0.7654	0.7677	0.3	0.7655,0.7667,0.7653	0.0,0.3,0.0
4	1.1111	1.1214	0.9	1.1138,1.1138,1.1136	0.2,0.2,0.2
5	1.4142	1.4291	1.0	1.4299,1.4232,1.4230	1.1,6.4,6.2
6	1.6629	1.6652	0.2	1.6878,1.6683,1.6760	1.5,0.3,0.8
7	1.8478	1.8488	0.1	1.9664,1.8686,1.8750	6.4,1.1,1.5
8	1.9616	1.9652	0.2	1.9876,1.9611,1.9638	1.3,0.0,0.1

(c) Same integrated input, same noise variance, deadbands

Figure 4.7.3: Comparison of Identifier Performance Using Optimal Input To Using PRBS for 8 Mass Slinky

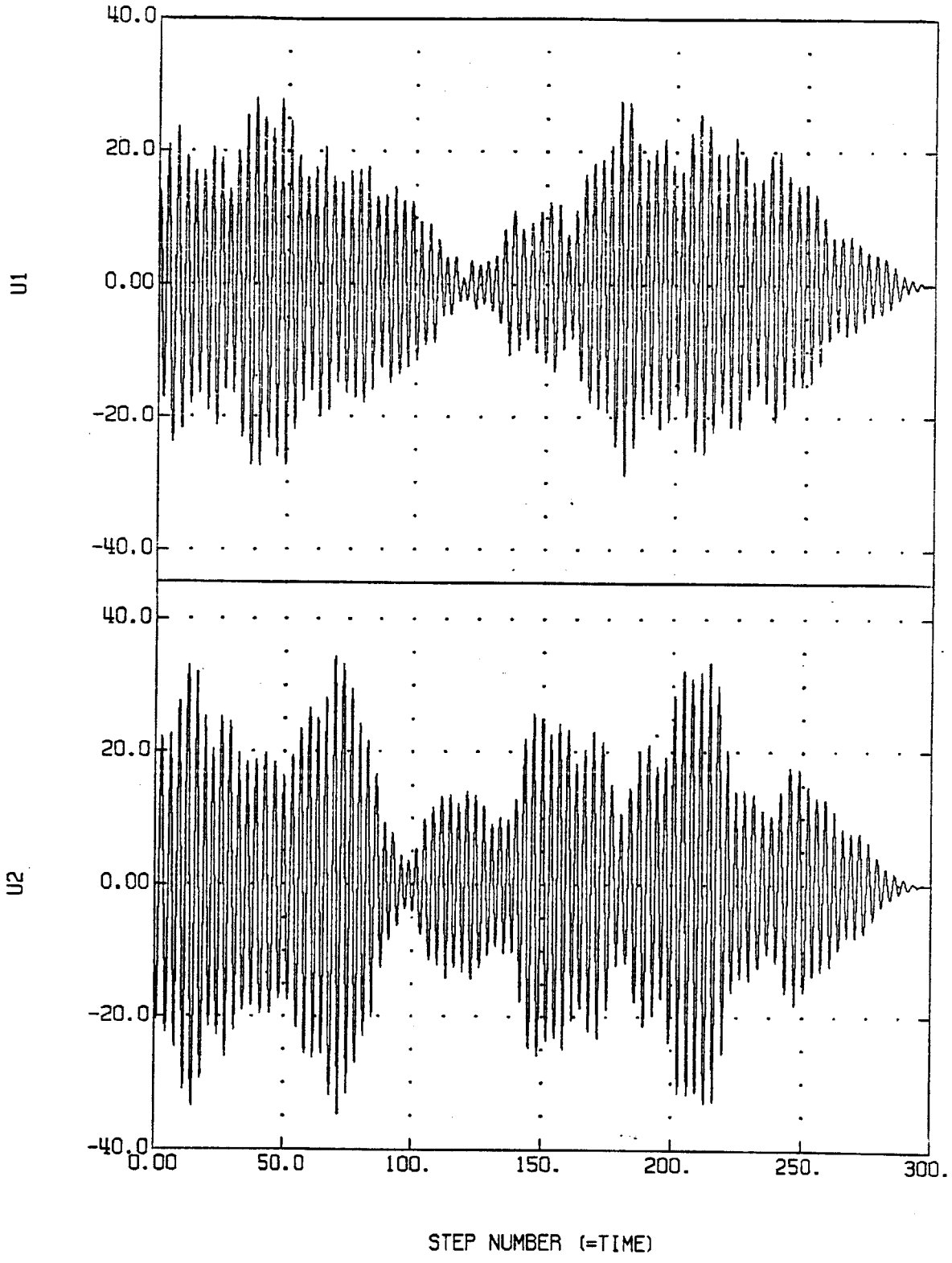


Figure 4.7.4 Optimal Input for 8 Mass Slinky

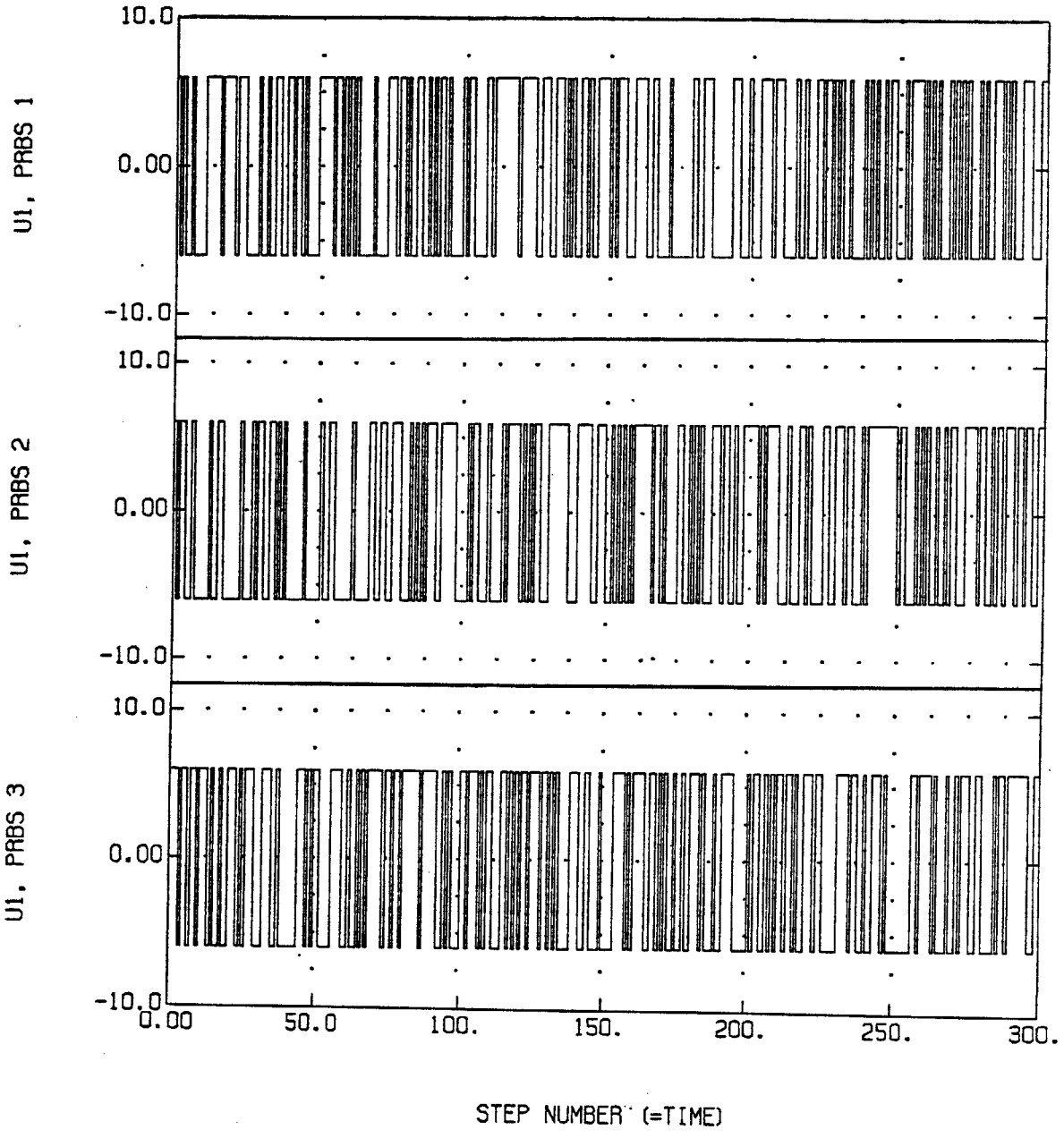


Figure 4.7.5 3 PRBS Inputs for Actuator 1 for 8 Mass Slinky

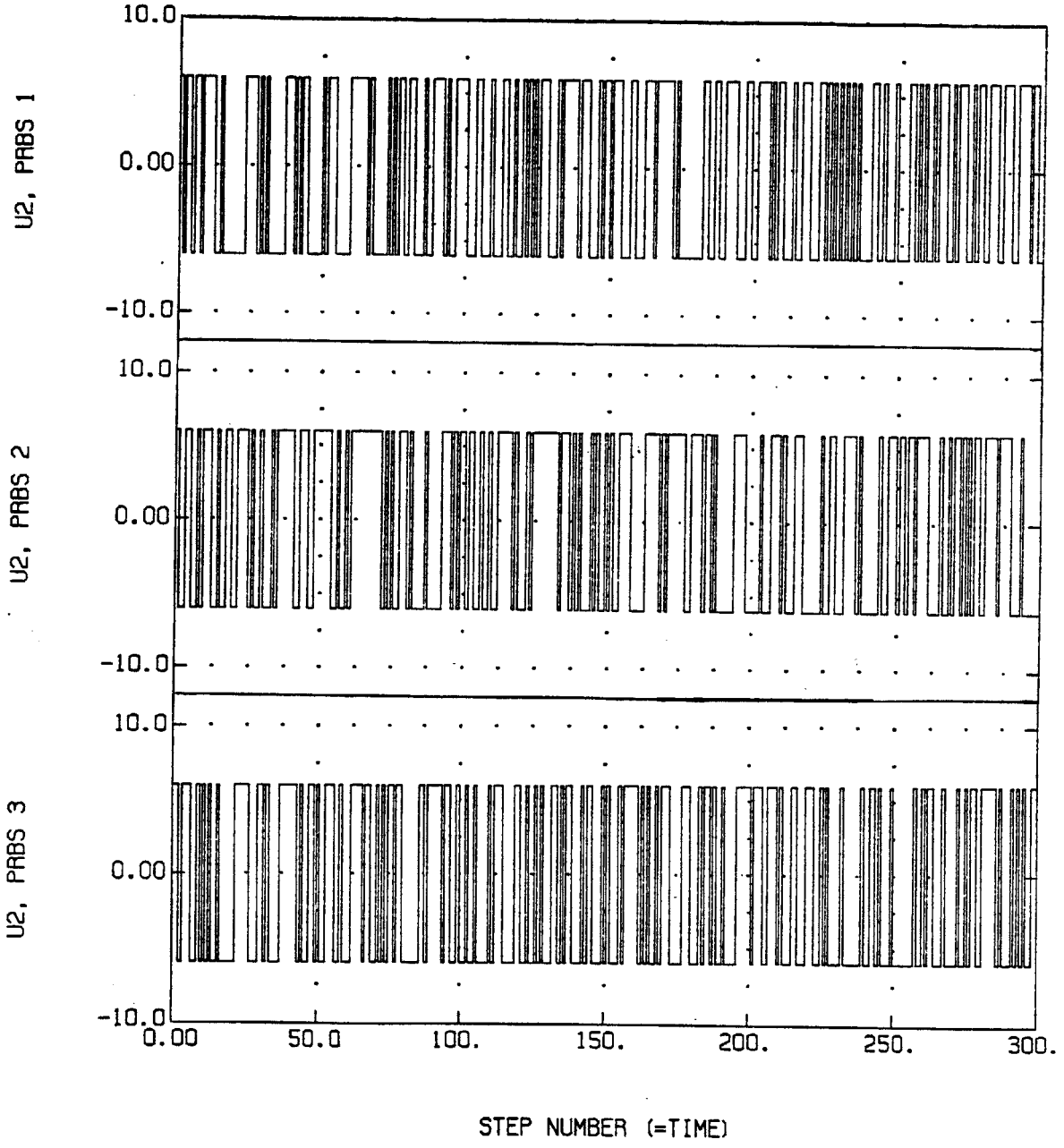


Figure 4.7.6 3 PRBS Inputs for Actuator 2 for 8 Mass Slinky

Mode	True Damped Frequency	Optimal Estimate	% Error	3 PRBS Estimates	% Error
1	0.0000	0.0023		0.0118,0.0003,0.0081	
2	0.3847	0.3864	0.4	0.3849,0.3850,0.3848	0.0,0.0,0.0
3	0.7575	0.7589	0.2	0.7584,0.7598,0.7582	0.1,0.3,0.1
4	1.1286	1.1513	2.0	1.1452,1.1343,1.1320	1.5,0.5,0.3
5	1.4181	1.4406	1.6	1.4575,1.4459,1.4546	2.8,2.0,2.6
6	1.7158	1.7567	2.4	1.8566,1.8665,1.8606	8.2,8.8,8.4
7	1.8551	1.8562	0.0	1.8959,1.8812,1.8786	2.2,1.4,1.3
8	2.0331	2.0380	0.2	2.0505,2.0580,2.0430	0.8,1.2,0.5

Figure 4.7.7: Comparison of Identifier Performance Using Optimal Input To Using PRBS for 8 Mass Slinky With Modelling Errors

Thus, Figure 4.7.1 illustrates three things: (1) the optimal input performs better than a PRBS, even when that PRBS has been carefully chosen, using the a priori model of the system, to perform well for that particular system (an 'intelligent' binary sequence); (2) while some PRBSs may perform better than the optimal input for large modelling errors, a random choice is more likely to come up with a PRBS which is less effective than the optimal input than one which is better; (3) the performance of the optimal input degrades in the presence of modelling errors, but so do many PRBSs.

4.8 Unknown Elements in Observation Matrix

The elements of the C matrix in equation 4.3.4 are combinations of sensor calibration factors and mode shapes. Usually, the calibration factors are known well and don't need to be experimentally measured. If the sensors and actuators are colocated, then the elements of the C matrix contain no information which can't be obtained by optimizing with respect to elements of the B matrix. If there is new mode shape information in the C matrix, then optimizing with respect to its elements may be desirable. Identifying unknown elements in C is a special case of limiting state amplitude, as shown by the simple example below:

$$\dot{\underline{x}} = A\underline{x} + Bu$$

$$y = [C_1 \; C_2]\underline{x}$$

$$\therefore \quad \dot{\underline{x}}_{C_1} = A\underline{x}_{C_1}$$

$$y_{C_1} = [1 \quad 0]\underline{x} + [C_1 \quad C_2]\underline{x}_{C_1}$$

But $\underline{x}_{C_1}(0) = [0 \quad 0]^T$ since $\underline{x}(0)$ is fixed:

$$\therefore \quad \underline{x}_{C_1}(t) = [0 \quad 0]^T \rightarrow y_{C_1} = [1 \quad 0]\underline{x}$$

Thus, maximizing the sensitivity of the output to an element of C is equivalent to maximizing the element of the state which it multiplies. This is the case of limiting state amplitude using a negative W (see section 4.6, Limiting State Amplitude).

4.9 MIMO LSS

Adding a thruster and sensor on mass 2 of the system in Figure 4.6.1 (as shown in Figure 4.9.1) makes this a MIMO structure. This adds the complexity of different modal content in the different inputs. Mass 2 is at the node of the first flex mode, so the optimal input from actuator 2 should have no mode 1 content. The optimal input from actuator 1 should have a lot of mode 1 content to make up for the lack of excitation from actuator 2.

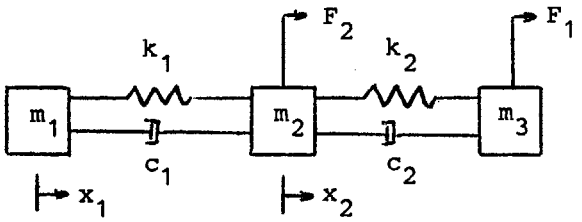


Figure 4.9.1: Three Mass, Two Input, Two Output Slinky

As expected, the optimal input on mass 2 is pure mode 2, and that on mass 3 is primarily mode 1. The frequencies used for this example are 1 and 4.3 rad/sec (damping ratio=.01 for both modes). 4.3 was used instead of 1.291 to make the frequency content of the signals easier to identify.

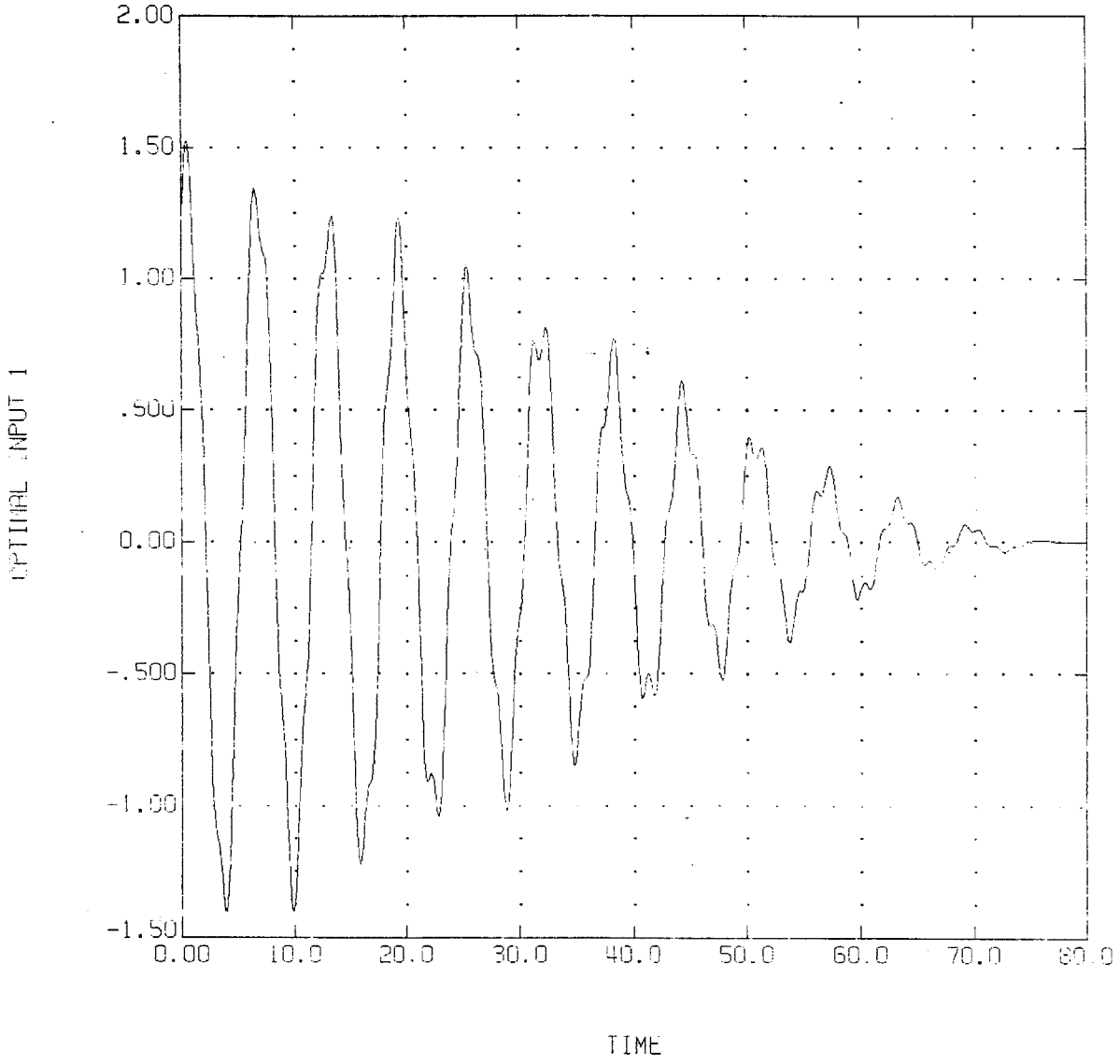


Figure 4.9.2 Optimizing With Respect to Two Frequencies for a MIMO LSS, Input 1

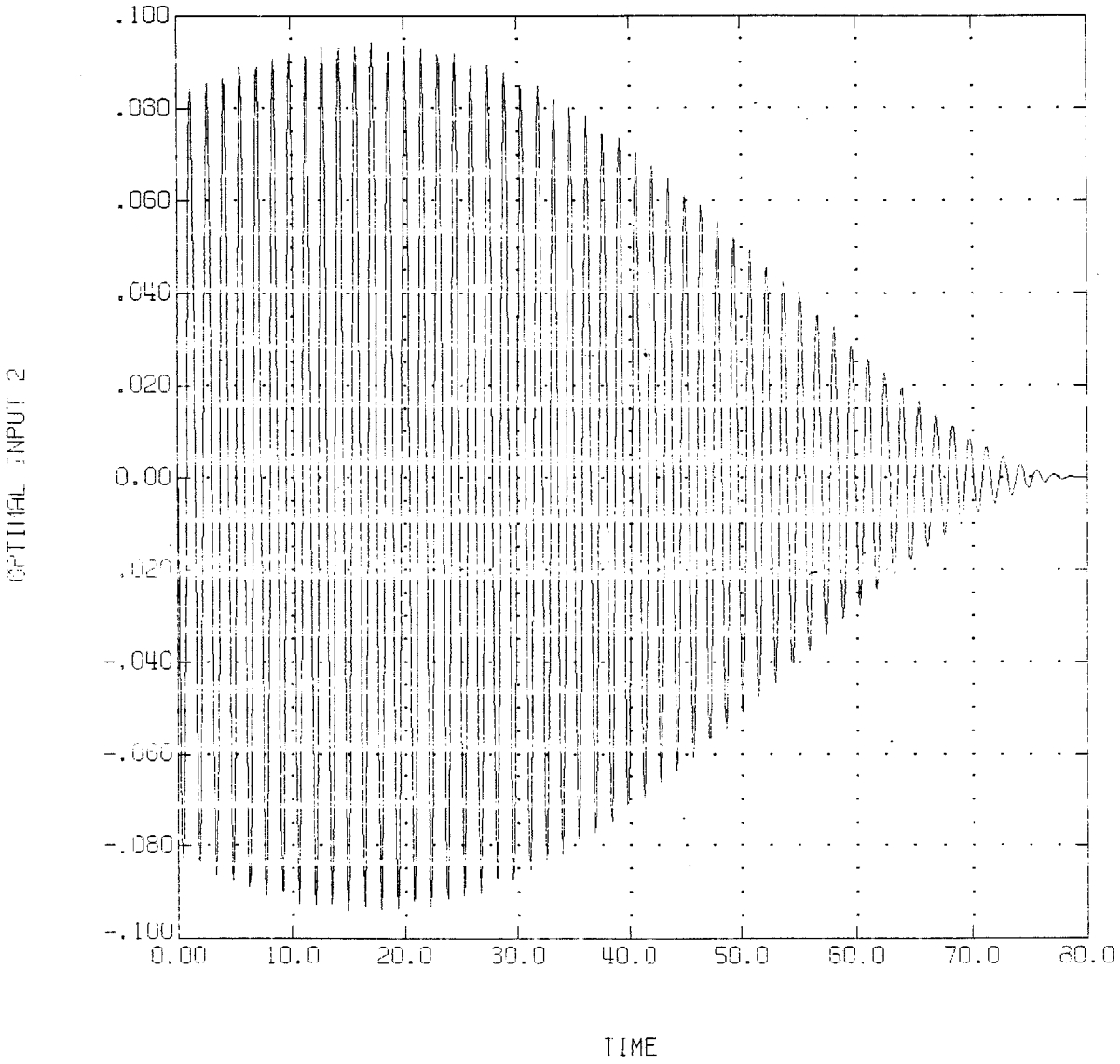


Figure 4.9.3 Optimizing With Respect to Two Frequencies for a MIMO LSS, Input 2

4.10 Programming Considerations

There are two potentially severe problems with the technique for finding optimal inputs (see Section 4.3): inversion of a large matrix and computation of independent eigenvectors for repeated eigenvalues. Both problems are solved by the same observation -- most of the elements of $\underline{\xi}(t)$ and $\underline{\lambda}(t)$ in equation (4.3.18) are zero or not needed to calculate the optimal input.

The easiest way to clarify this is to work through the equations for a low dimensional example. (To save space, 'X' is used to indicate non-zero elements in the larger matrices.) Consider a SISO system with 2 flexible modes, and optimize with respect to the damping ratio of the first mode, weighted by 1:

$$(4.3.1) \rightarrow \dot{\underline{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & -2\zeta_0\omega_0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_1^2 & -2\zeta_1\omega_1 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ b_{21} \\ 0 \\ b_{41} \end{bmatrix} u$$

$$(4.3.2) \rightarrow y = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \end{bmatrix}$$

$$(4.3.5) \rightarrow \dot{\underline{\xi}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & X & X & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & X & 0 & 0 & X & X & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & X & X \end{bmatrix} \underline{\xi} + \begin{bmatrix} 0 \\ X \\ 0 \\ X \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

$$(4.3.5) \rightarrow \begin{bmatrix} \dot{\underline{\xi}} \\ \dot{\underline{\lambda}} \end{bmatrix} = \left[\begin{array}{cccccccc|cccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & 0 & X & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & X & X & 0 & 0 & 0 & 0 & 0 & X & 0 & X & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & X & 0 & 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & X & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X & X & X & X & 0 & 0 & 0 & 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & 0 & X & X & X & X & 0 & 0 & 0 & 0 & -1 & X & 0 & 0 \\ 0 & 0 & 0 & 0 & X & X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X \\ 0 & 0 & 0 & 0 & X & X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & -1 & X \end{array}\right] \begin{bmatrix} \underline{\xi} \\ \underline{\lambda} \end{bmatrix}$$

The boundary conditions in equation 4.3.15 give ξ_7 and ξ_8 zero at t=0 and λ_3 and λ_4 zero at t=T. The equations at the bottom of the previous page show that these variables are part of unforced, uncoupled second order systems. With zero boundary conditions, they will always be zero. Further examination shows that ξ_3, ξ_4, λ_7, and λ_8, while possibly non-zero because they are driven by other variables, do not in turn drive any other variables, so they are not needed to solve the problem. Eliminating these 8 variables:

$$\begin{bmatrix} \dot{\underline{\xi}} \\ \dot{\underline{\lambda}} \end{bmatrix} = \left[\begin{array}{cccc|cccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & 0 & 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & X & X & X & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & X & 0 & X \\ 0 & 0 & X & X & 0 & 0 & 0 & X \\ 0 & 0 & X & X & 0 & 0 & -1 & X \end{array}\right] \begin{bmatrix} \underline{\xi} \\ \underline{\lambda} \end{bmatrix}$$

The resulting system is half the original size and has no repeated eigenvalues. (As demonstrated in Figure 4.6.2, the eigenvalues will all approach $-\zeta_0\ \omega_0 \mp \omega_0 \sqrt{1 - \zeta_0^2}$ as the experiment time lengthens, but they are all distinct.)

If at least one parameter from every mode is included in the optimization, then no variables can be eliminated, but there are no repeated eigenvalues. For every parameter in the optimization, the number of variables which need to be kept is 2 for the sensitivity for that parameter, 2 for its costate, plus 2 for each mode involved and 2 for each of those costates. The total ranges from 4*r + 4 to 8*r, depending on the number of modes involved (where r = number of parameters in the optimization).

If the time history of any of the deleted variables is of interest, it can be easily recovered by saving the appropriate rows of the eigenvector matrix in equation 4.3.18. The reduced size system is used to calculate the β_i in the columns of Φ which correspond to nonzero values of γ_i, and to calculate those non-zero values.

Chapter 5: Results for a Typical LSS

All the examples used in the previous chapters were slinkys, which were useful for learning the characteristics of the RLLS algorithm and for comparing different algorithms. A system which more closely resembles a typical LSS is desirable for final verification. Also, the data from such a model is more easily compared to data presented by other researchers. This chapter presents such a model and applies the identification algorithms of Chapters 2 and 3 and the optimal input from Chapter 4 to it.

5.1 Dual Keel Space Station Model

As a result of NASA's preliminary work on designing a space station, models of a likely space station have been developed. One of these models -- the "dual keel" -- was chosen as the example of a typical LSS. Figure 5.1.1 is a drawing of this LSS. One of the problems in assembling such a large structure with current technology is that it cannot be assembled on one trip to orbit. The configuration will be different after each trip, with orbital maintenance required between trips. Identifying the structure characteristics as the structure grows will be necessary to safely and effectively control the station between assemblies.

The particular stage used in this chapter would be typical of the station configuration after 4 assembly flights. This corresponds to Figure 5.1.1 without the modules in the center and without the 4 bays and the instrument storage. The resulting structure has a mass of 34,000 kg. It measures 405 meters from tip to tip of the booms carrying the solar arrays. The dual keel box measures 306 by 108 meters. The simulated measurements are generated by a NASTRAN model which divided the station into 157 nodes as shown in Figure 5.1.2. (Note that the structure for the central modules is in place even though the modules aren't.) Each gap shown in the figure is a node. The numbered nodes are where actuators and/or sensors were placed. (The node numbering is that used in the NASTRAN output. Adjacent nodes do not always have sequential numbers.)

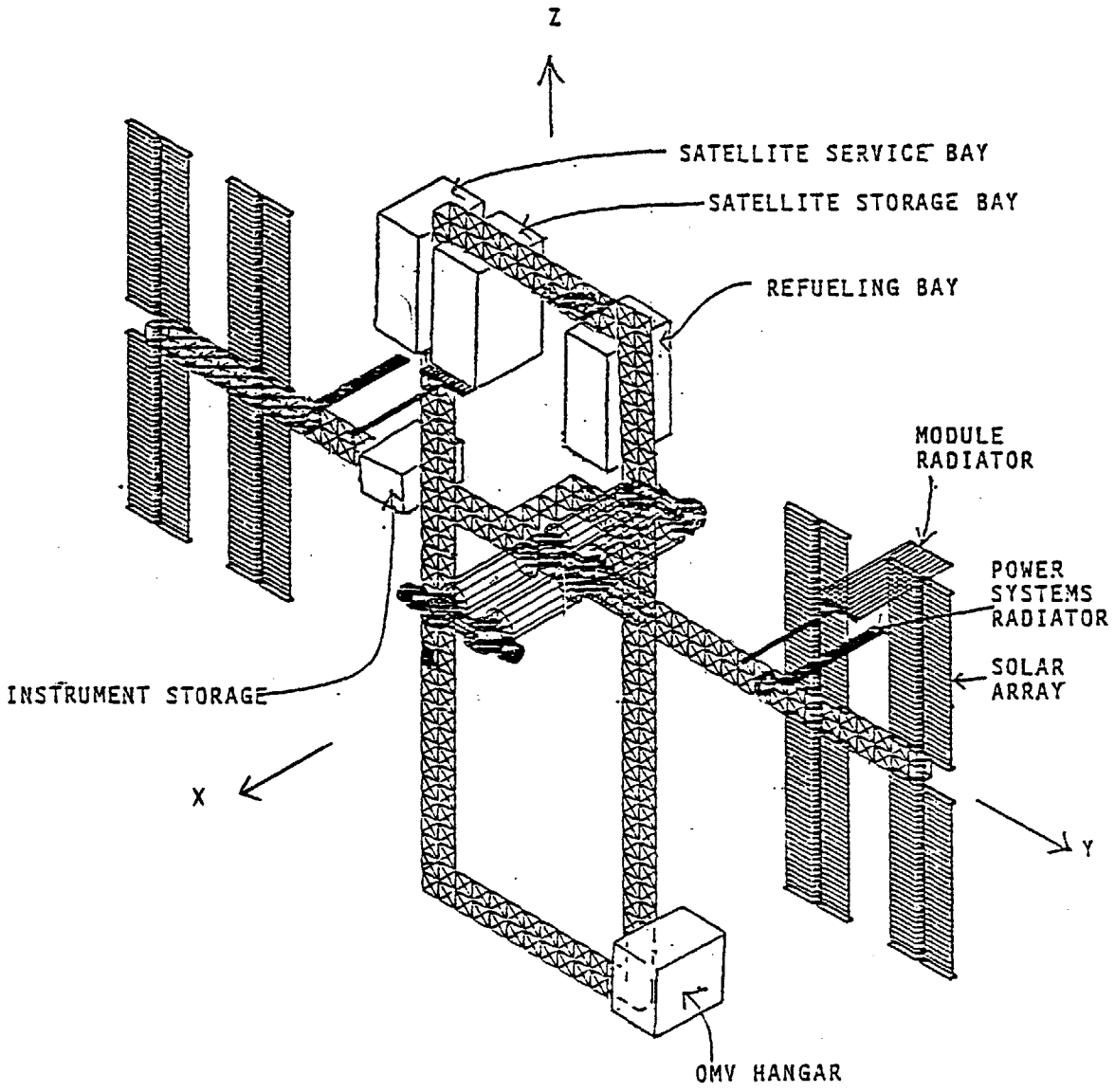
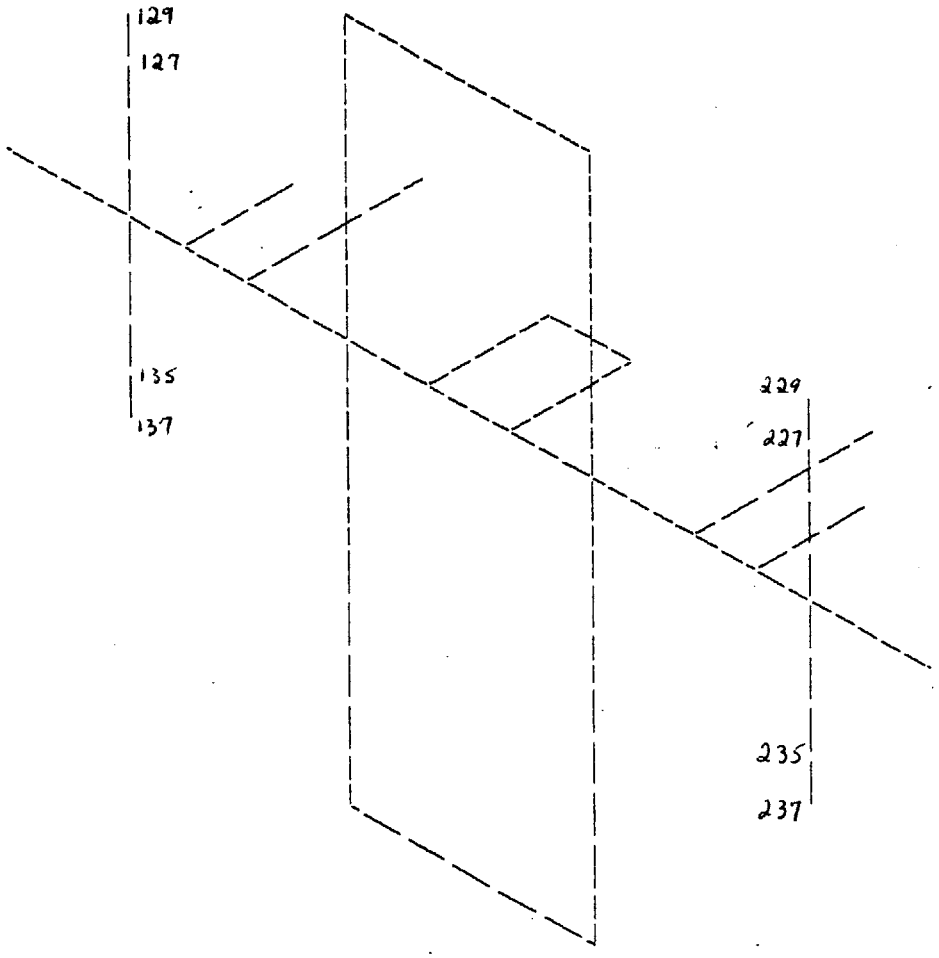


Figure 5.1.1: Dual Keel Space Station



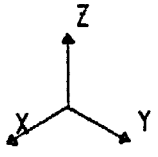


Figure 5.1.2: NASTRAN Model of the Dual Keel Space Station After 4 Assembly Flights

Model Size

The NASTRAN model included 6 rigid body modes and 24 flexible modes ranging from 1.3 rad/sec (.21 Hz) to 8.1 rad/sec (1.3 Hz), as listed in the following table.

Mode	1	2	3	4	5	6	7	8
Frequency	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.2977	1.3543
Mode	9	10	11	12	13	14	15	16
Frequency	1.3568	1.3734	1.3771	1.3980	1.4045	1.4257	2.3855	2.5701
Mode	17	18	19	20	21	22	23	24
Frequency	2.8274	3.1788	3.2922	3.3652	3.9364	4.1165	4.7312	4.7837
Mode	25	26	27	28	29	30		
Frequency	6.1456	6.5253	7.8914	8.0491	8.0507	8.0994		

Table 5.1.1: Modal Frequencies in NASTRAN Model

The original intention was to use all 30 modes. Unfortunately, this turned out to be not feasible due to the close spacing of the first 8 flexible modes.

As frequencies get more closely spaced, finite accuracy errors begin to cause problems. Finite accuracy errors are identified by: (1) failure of the algorithm to converge exactly to the known model in a finite number of steps in the absence of noise; (2) significant changes in the results when the numerical precision of the computer code is changed. Both of these were observed in the simulations.

This causes two problems. The first problem is that sampling rate becomes very important. A sampling rate between the Nyquist frequency and twice the Nyquist frequency of the closely spaced modes is best for minimizing the effects of finite accuracy error[24]. To prevent aliasing, this limits the modes included in the model to those in that cluster or a few nearby clusters, depending on the spacing. The second problem is that the round off errors incurred by using the larger matrices required by the identifier for large models become more significant.

The result is that a smaller model had to be used. One way to make the system look smaller is to use a band pass filter to isolate clusters of modes. This was simulated by using the first eight

flexible modes as the station model. Even so, the close spacing of the eight modes means that the identification problem is more difficult than for the eight mode cases examined in Chapters 2 and 3.

Unfortunately, this eight mode model is a poor test for the optimal input algorithm. When the modes are so close together, a square wave at the center frequency of the closely spaced modes gives results very similar to the optimal input. The optimal input algorithm would be more valuable on a LSS such as the Langley Mast flight experiment[2] (first five flexible modes at .97, 1.3, 9.1, 9.2, and 17.2 rad/sec) or the Galileo spacecraft[7] (first five flexible modes at 69.6, 70.3, 93.8, 94.7, and 102.7 rad/sec).

5.2 Simulation Environment

Noise sources

In addition to the measurement noise used in the examples in the previous chapters, system disturbance is included in the form of gravity gradient and atmospheric drag.

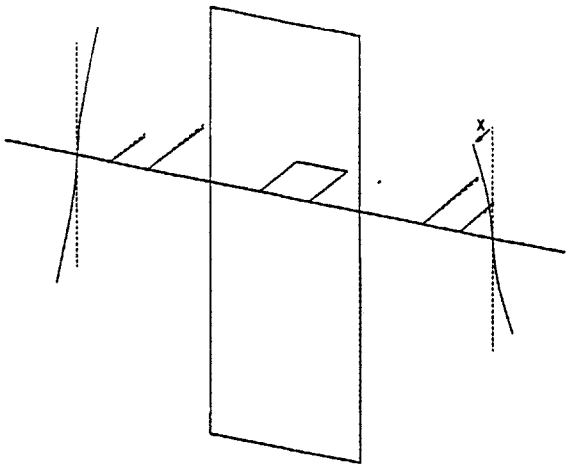
The station is oriented with the z axis (the long direction of the dual keel box, see Figure 5.1.2) pointed toward the center of the earth. This orientation is stabilized by gravity gradient forces. The inputs used to excite the structure for identification cause variations from this stable position, generating unaccounted-for gravity gradient forces on the structure. The gravity model assumes a homogeneous, spherical earth with the station in a circular, 500 km orbit.

Since the time scale of the experiments is short (on the order of 5 to 10 minutes), the atmospheric model is a constant density of 10^{-12} kg/m^3. The drag is almost entirely from the solar cell panels.

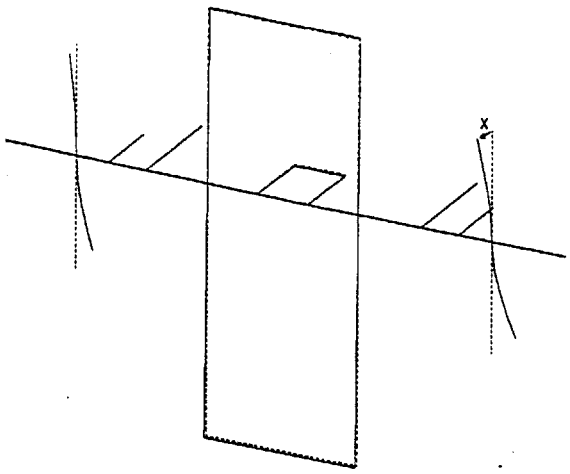
The measurement noise is the same as in previous examples i.e. Gaussian white noise generated by a random noise generator and added to the measurements.

Sensors and Actuators

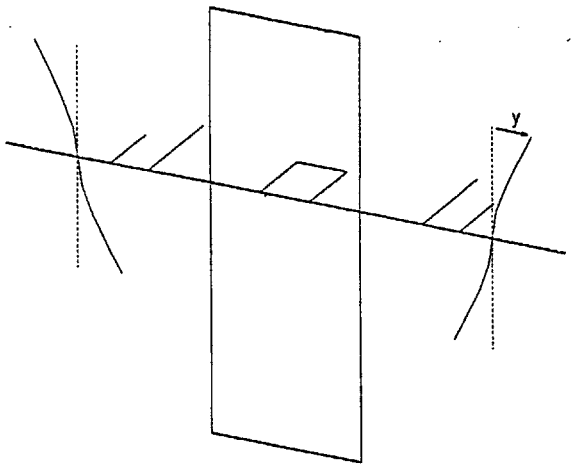
Since the clusters of flexible modes are primarily due to solar cell panel motion, the sensors and actuators were placed on the panels. There were four actuators, 2 firing in the x direction at nodes 129 and 137, and 2 in y at the same nodes (see figure 5.1.2).



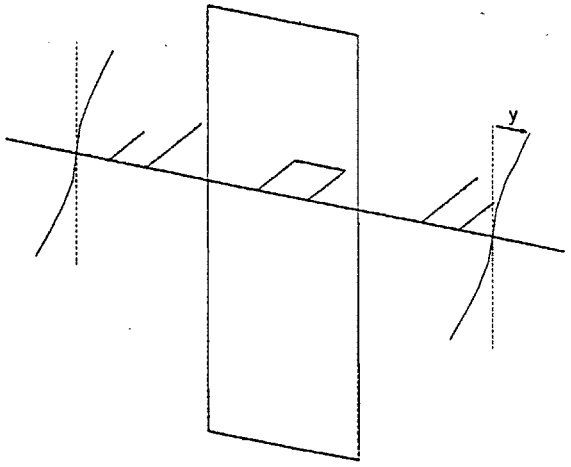
Mode 7, Flex mode 1



Mode 8, Flex mode 2

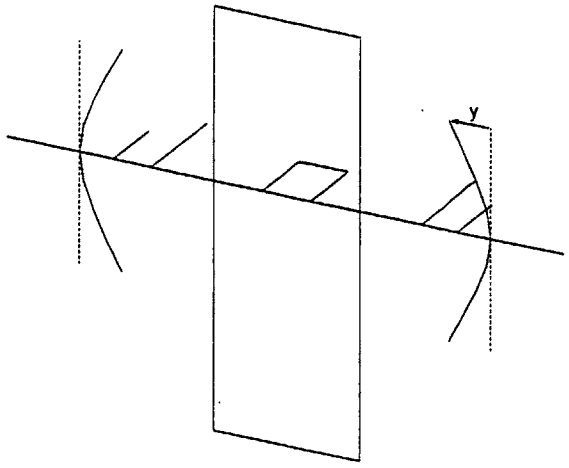


Mode 9, Flex mode 3

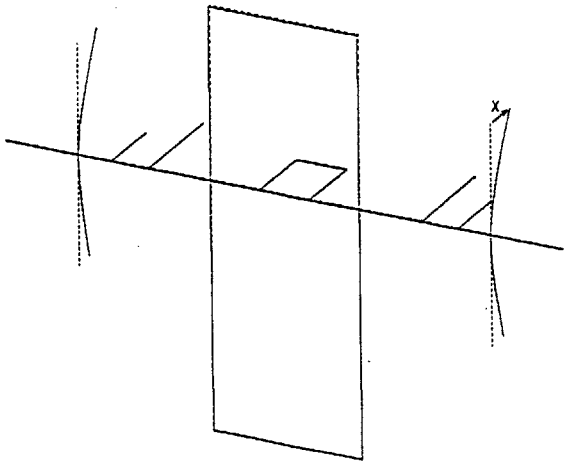


Mode 10, Flex mode 4

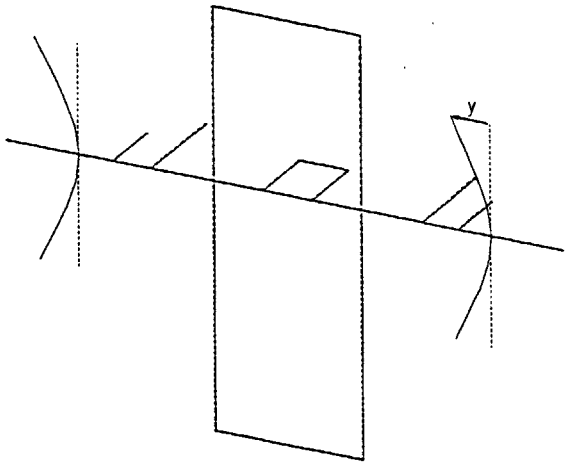
Figure 5.1.3: Mode Shapes for Space Station, First Four Flexible Modes



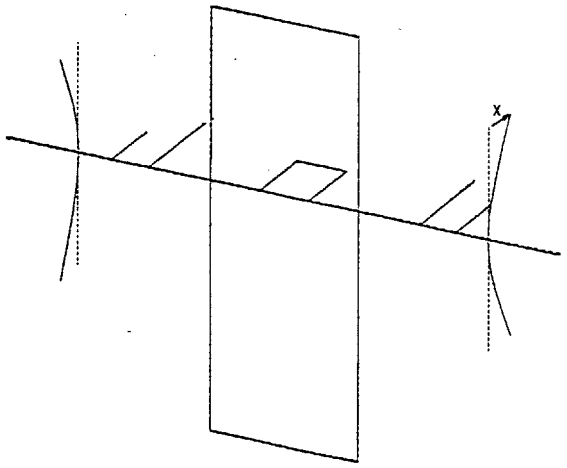
Mode 11, Flex mode 5



Mode 12, Flex mode 6



Mode 13, Flex mode 7



Mode 14, Flex mode 8

Figure 5.1.4: Mode Shapes for Space Station, Second Four Flexible Modes

two combinations of sensors were used: (1) four position sensors, 2 measuring inertial displacement in the x direction at nodes 129 and 135, and 2 measuring inertial displacement in the y direction at nodes 127 and 137; (2) those four, plus 2 measuring inertial y at nodes 229 and 235 and 2 measuring inertial x at nodes 227 and 237.

5.3 Results

The optimal input for the 4 thrusters is shown in Figures 5.3.1 and 5.3.2. The input was calculated by optimizing with respect to frequency with weighting factors of .01, .02, .04, .0065, .01575, .0179, .004, .003 on the eight flexible modes. This choice approximately equalizes the integrated sensitivity of the outputs with respect to each of the eight mode frequencies. The sample-and-hold input used in the simulation is shown in Figures 5.3.3 and 5.3.4. Note the strong aliasing of the signal for this case, where the sample rate is close to the dominant frequency of the signal. The input for the suboptimal identifier is the optimal input repeated four times, as shown in Figure 5.3.5 for actuators 1 and 2.

The first case studied was the four sensor case. When the same SNR (100) which was used in the Chapters 2 and 3 examples was used, only four of the eight modes were correctly identified by the optimal identifier, as shown in Table 5.3.1. The performance improves markedly with SNR. In the Table 5.3.2 results, the SNR has been increased to 1000. While the damping ratio estimates are poor, all eight frequencies were identified to within 4%. The suboptimal identifier, which had one-quarter the number of computations per cycle, also identified all eight frequencies to within 4%. (The suboptimal identifier performed better than the optimal in terms of frequency error (2% maximum error) for this particular run. The damping error was worse, however.) The identified mode shapes for the optimal input case, SNR = 1000, is shown in Figures 5.3.6-7.

As studied in reference 12, for closely spaced modes, identification accuracy can be improved by increasing the number of sensors to equal the number of closely spaced modes. As shown in Table 5.3.3, using 8 sensors gives better estimates at a SNR of 100 than 4

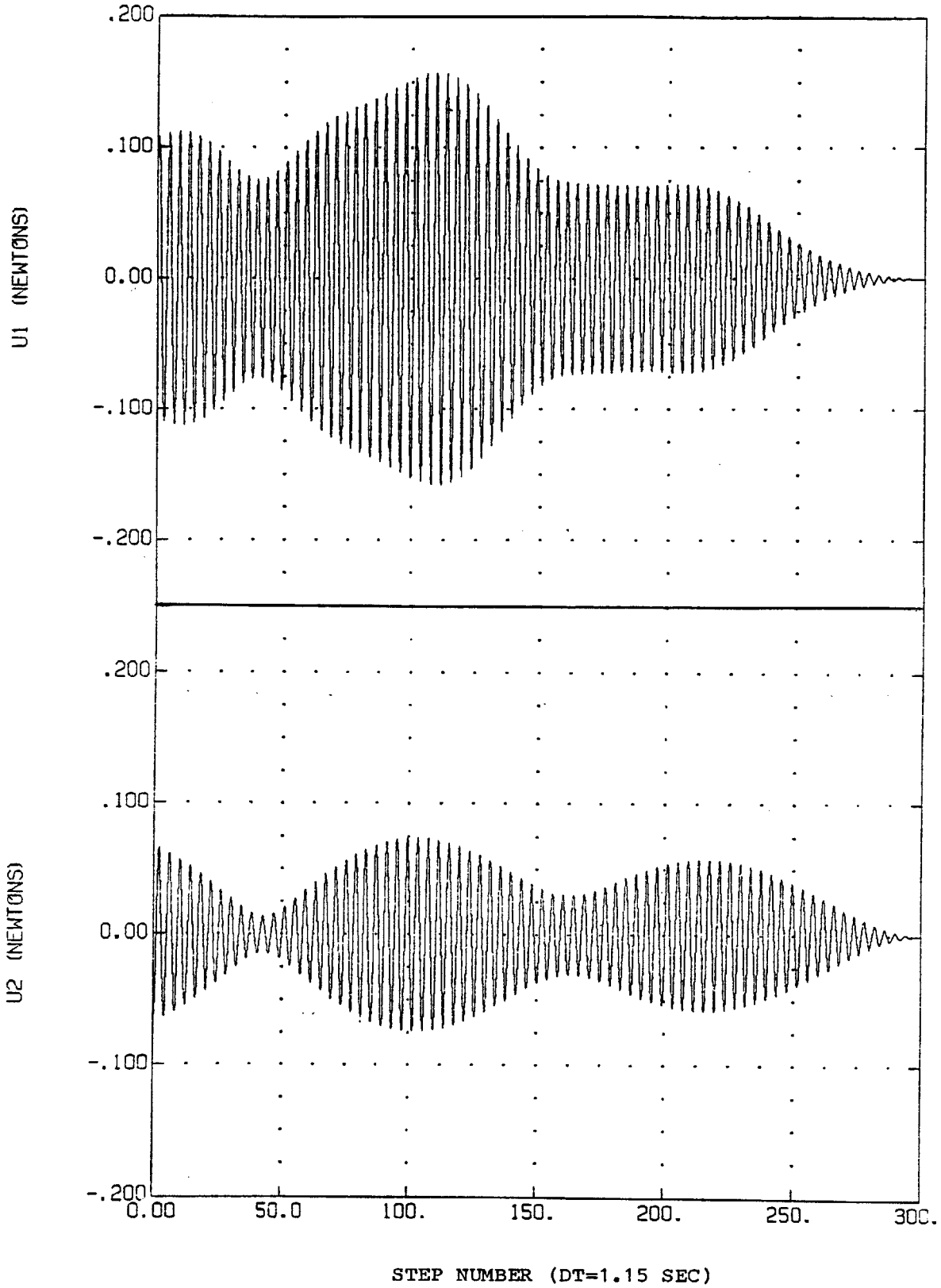


Figure 5.3.1: Optimal Input for Actuators 1 and 2

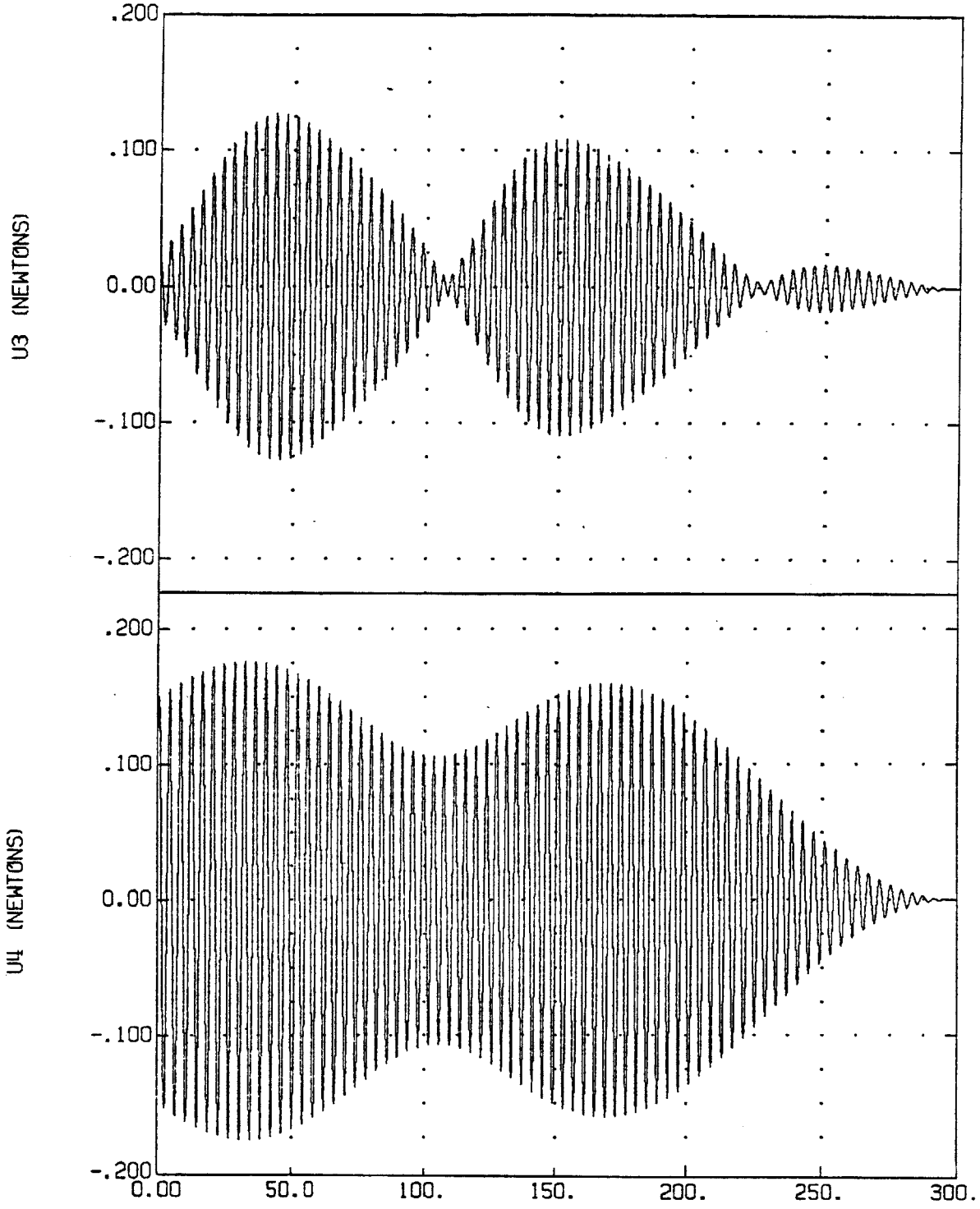


Figure 5.3.2: Optimal Input for Actuators 3 and 4

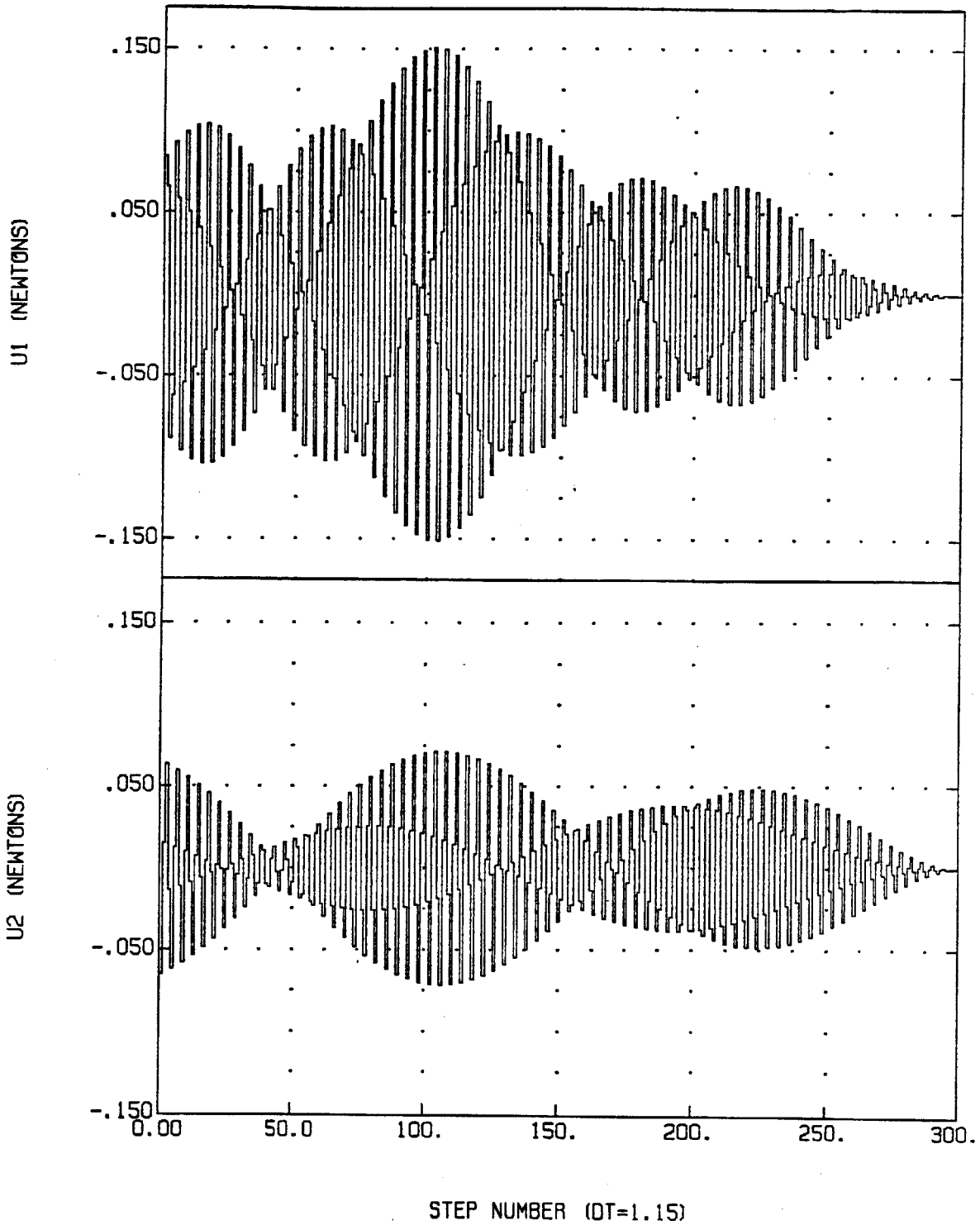


Figure 5.3.3: Sample-and-Hold Input for Actuators 1 and 2

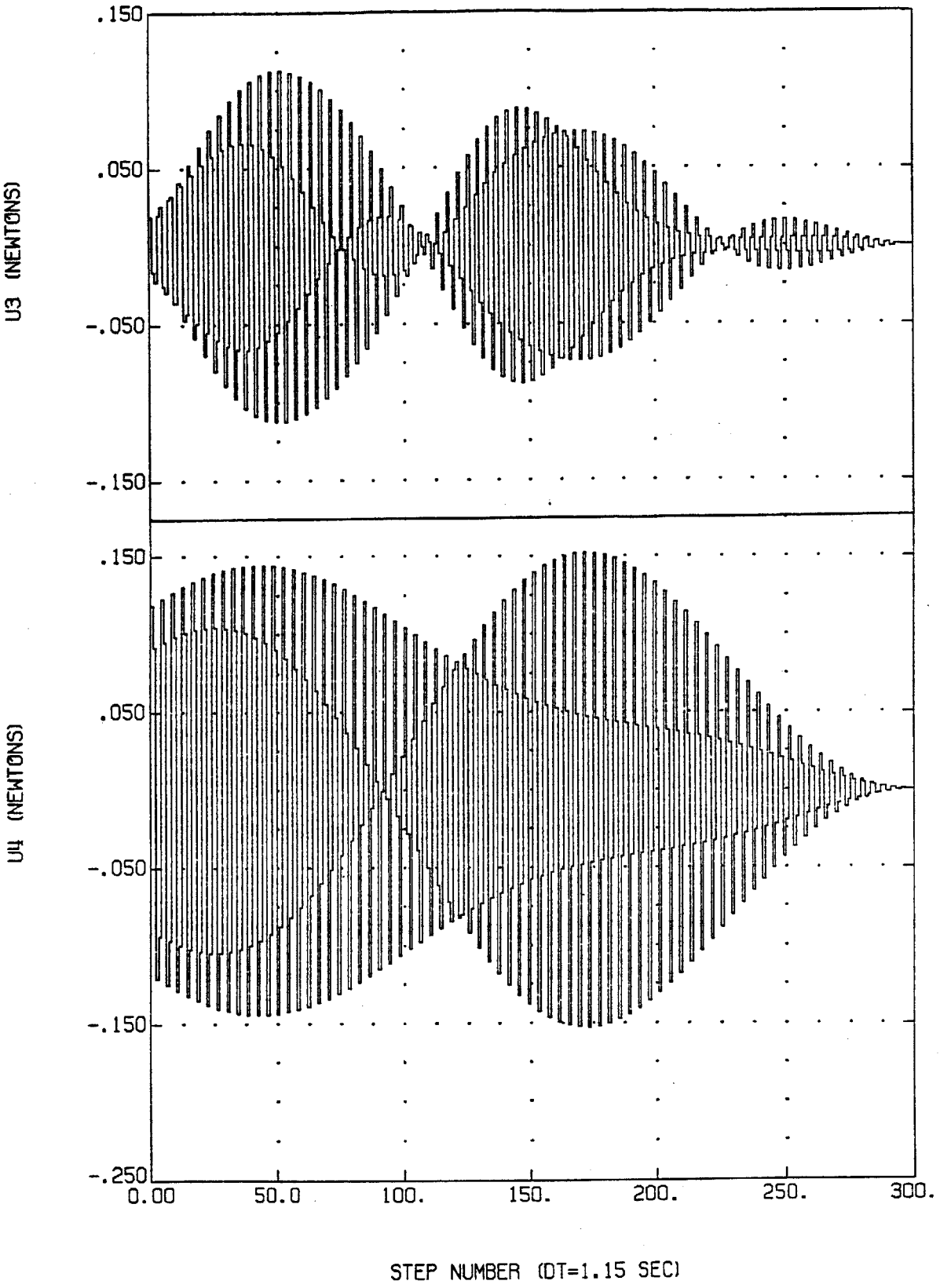


Figure 5.3.4: Sample-and-Hold Input for Actuators 3 and 4

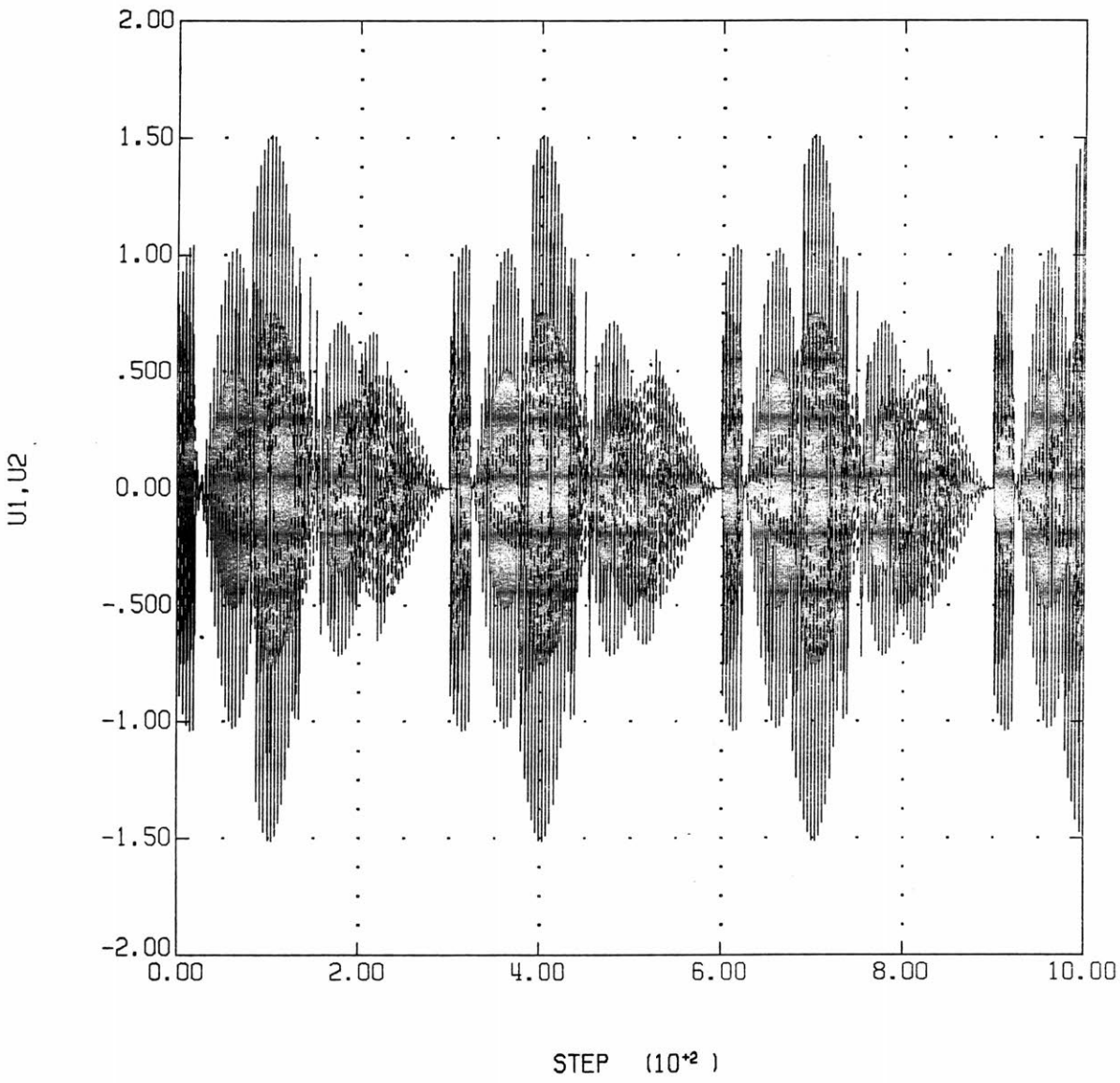


Figure 5.3.5: Input for Suboptimal Identifier, Actuators 1 and 2

Mode	True Damped Frequency	Optimal Estimate
1	1.2976	1.3202
2	1.3542	*lost
3	1.3567	1.3650
4	1.3733	lost
5	1.3770	1.3811
6	1.3879	lost
7	1.4044	lost
8	1.4256	1.4243

(a) Identified Frequencies

Mode	Damping Ratio	Optimal Estimate
1	.01	.0463
2	.01	-
3	.01	.0180
4	.01	-
5	.01	.0185
6	.01	-
7	.01	-
8	.01	.0056

(b) Identified Damping Ratios

Table 5.3.1: Identified Frequencies and Damping Ratios, 4 Sensors, SNR = 100

*lost means identified as a pair of real roots instead of a complex conjugate pair or error in frequency estimate greater than 50%

Mode	True Damped Frequency	Optimal Estimate	Suboptimal Estimate
1	1.2976	1.2978	1.2974
2	1.3542	1.3515	1.3620
3	1.3567	1.3634	1.3641
4	1.3733	1.4258	1.4057
5	1.3770	1.3802	1.3763
6	1.3879	1.3967	1.3927
7	1.4044	1.3917	1.4114
8	1.4256	1.4250	1.4214

(a) Identified Frequencies

Mode	Damping Ratio	Optimal Estimate	Suboptimal Estimate
1	.01	.0105	.0217
2	.01	.0103	.0032
3	.01	.0132	.0116
4	.01	.2203	.8049
5	.01	.0169	.0157
6	.01	.0119	.0146
7	.01	.0491	.1065
8	.01	.0145	.0182

(b) Identified Damping Ratios

Table 5.3.2: Identified Frequencies and Damping Ratios, 4 Sensors, SNR = 1000

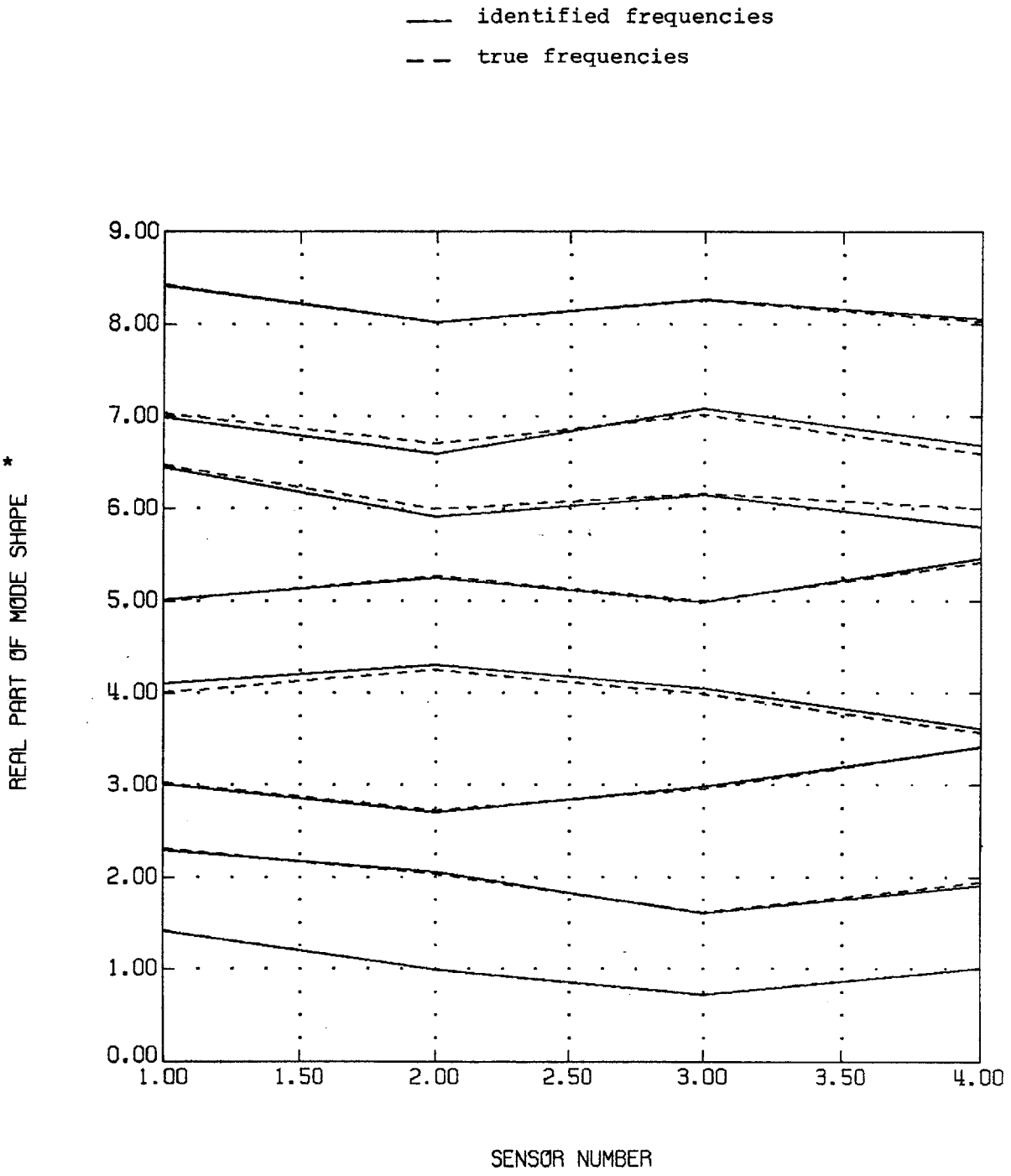


Figure 5.3.6: Real Part of Identified Mode Shapes,
4 Sensors, SNR = 1000

*Integers are mode number; deflections about 0 for each integer are the mode shape

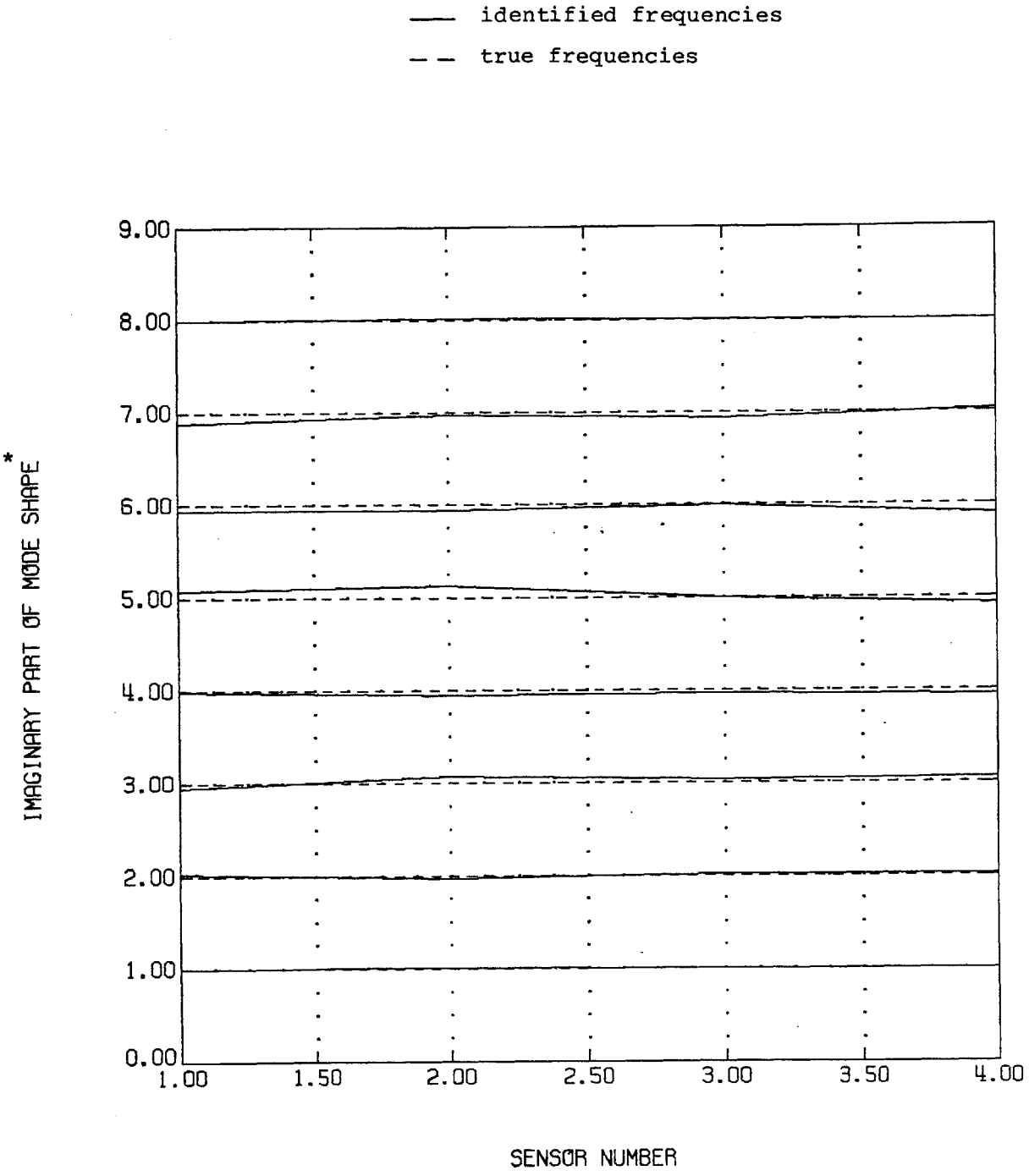


Figure 5.3.7: Imaginary Part of Identified Mode Shapes, 4 Sensors, SNR = 1000

*Integers are mode number; deflections about 0 for each integer are the mode shape

Mode	True Damped Frequency	Optimal Estimate	Suboptimal Estimate
1	1.2976	1.2958	1.2991
2	1.3542	1.3547	1.3521
3	1.3567	1.3612	1.3545
4	1.3733	1.3616	1.3748
5	1.3770	1.3760	1.3838
6	1.3879	1.3980	1.3959
7	1.4044	1.3991	1.4057
8	1.4256	1.4224	1.4265

(a) Identified Frequencies

Mode	Damping Ratio	Optimal Estimate	Suboptimal Estimate
1	.01	.0107	.0107
2	.01	.0126	.0003
3	.01	.0241	.0105
4	.01	.0130	.0093
5	.01	.0102	.0084
6	.01	.0117	.0127
7	.01	.0175	.0148
8	.01	.0127	.0094

(b) Identified Damping Ratios

Table 5.3.3: Identified Frequencies and Damping Ratios, 8 Sensors, SNR = 100

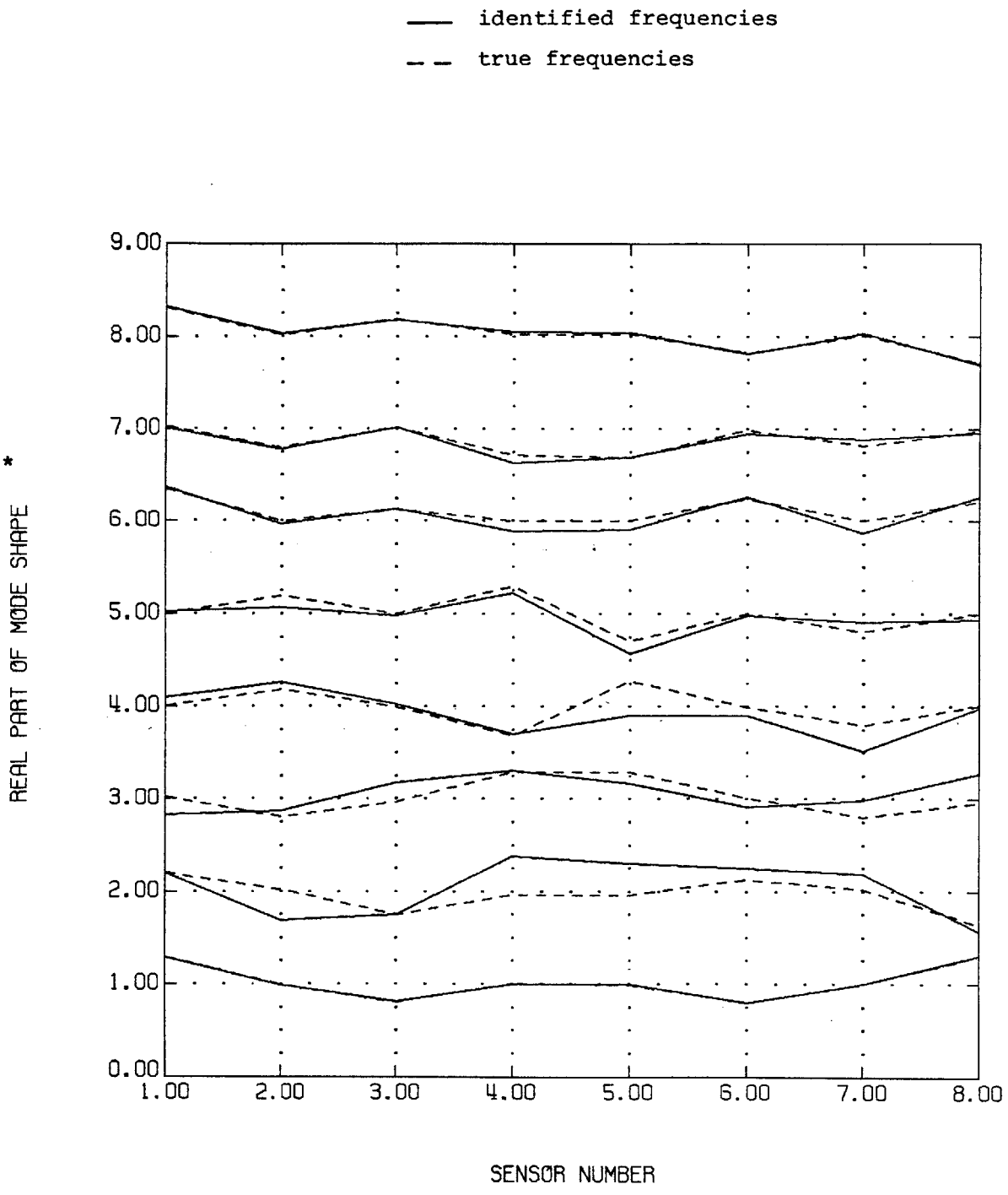


Figure 5.3.8: Real Part of Identified Mode Shapes,
8 Sensors, SNR = 100

*Integers are mode number; deflections about 0 for each integer are the mode shape

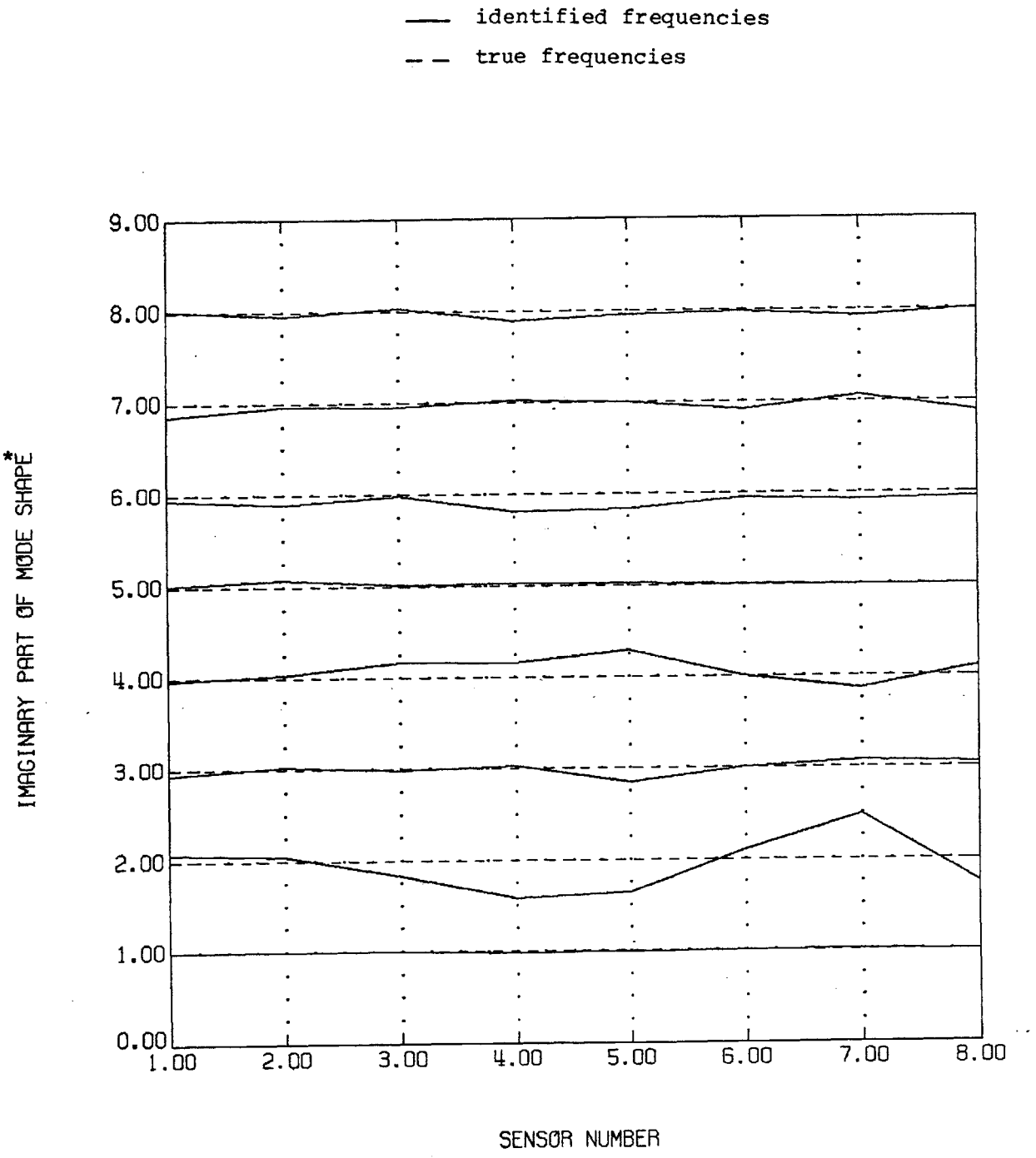


Figure 5.3.9: Imaginary Part of Identified Mode Shapes, 8 Sensors, SNR = 100

*Integers are mode number; deflections about 0 for each integer are the mode shape

significant improvement over the 4 sensor case. The identified mode shapes are shown in Figures 5.3.8-9. (The true mode shapes look different from the previous case because there are 8 sensors. The first halves of the true mode shapes in Figures 5.3.8-9 are the same as the true mode shapes in Figures 5.3.6-7.)

As mentioned earlier, a good PRBS input is easy to pick for the closely spaced modes case. Such an input is shown in Figure 5.3.10. It is a square wave at the center frequency of the closely spaced modes. The inputs from actuators 1 and 2 have the same time history, as do inputs from actuators 3 and 4. The phasing between these two pairs was explored a little, but it appeared to have no significant effect on the results. The identification results are presented in Table 5.3.4 for two phasing patterns. The results are similar to those for the optimal input i.e. the maximum frequency error is 1.5%.

Is this still true in the presence of modelling errors? Modelling errors were simulated by moving the center frequency of the clustered modes down 10% and spreading the modes evenly over the interval. The optimal input was the same as the one used in the previous paragraphs. The identifier estimates are shown in Table 5.3.5. Two cases are presented. For the first case, the simulation was exactly the same as the no modelling errors case except the frequencies were changed. This resulted in a much lower SNR (=30) because the structure was not as strongly excited. If this happened in practice, the most likely result is that the operator would increase the input until the LSS began to hit the deadbands. The results for that procedure are also shown in Table 5.3.5, and they are much better. In fact, they are better than the no modelling errors case for the same SNR (=100). This is due to the modes being less closely spaced. That helps the identifier more than the modelling error hurts it. The identified mode shapes are shown in Figures 5.3.11-5.3.13.

To summarize the results, the conclusions drawn in previous chapters (i.e. (1) the optimal algorithm can be used to accurately identify a LSS; (2) the suboptimal algorithm provides estimates only slightly worse than the optimal; (3) using the optimal input improves identifier performance) have been demonstrated on a more complicated model. The overall performance was degraded due to the extremely close spacing of the flexible modes.

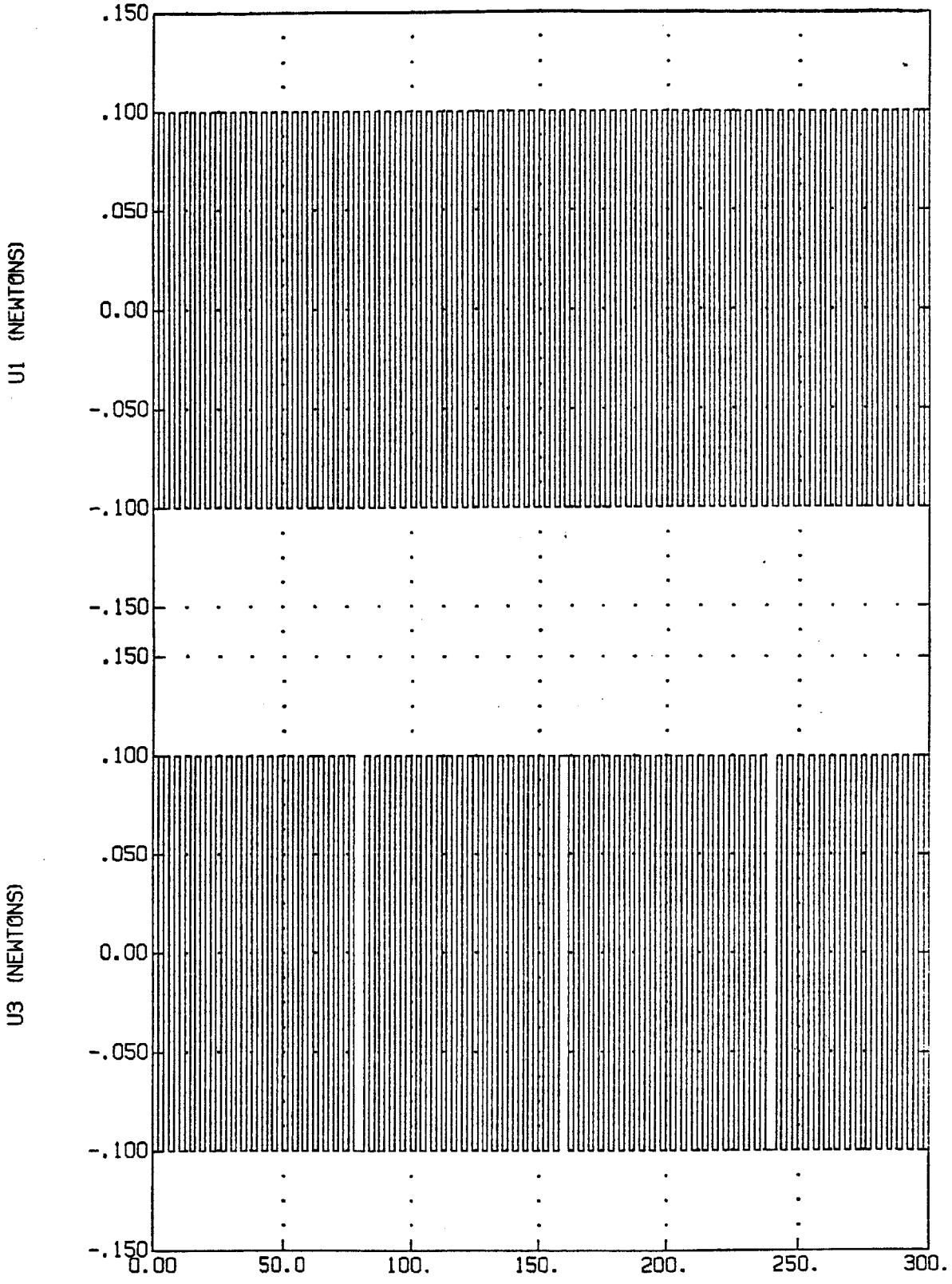


Figure 5.3.10: 'Good' PRBS Input for Actuators 1 (Same as 2) and 3 (Same as 4)

Mode	True Damped Frequency	*PRBS 1 Estimate	*PRBS 2 Estimate
1	1.2976	1.2984	1.2982
2	1.3542	1.3707	1.3584
3	1.3567	1.3557	1.3576
4	1.3733	1.3636	1.3768
5	1.3770	1.3768	1.3775
6	1.3879	1.3693	1.3589
7	1.4044	1.4020	1.4020
8	1.4256	1.4242	1.4245

(a) Identified Frequencies

Mode	Damping Ratio	PRBS 1 Estimate	PRBS 2 Estimate
1	.01	.0098	.0101
2	.01	.0096	.0101
3	.01	.0104	.0112
4	.01	.1126	.0115
5	.01	.0108	.0099
6	.01	.0468	.0859
7	.01	.0107	.0110
8	.01	.0102	.0110

(b) Identified Damping Ratios

Table 5.3.4: Identified Frequencies and Damping Ratios, PRBS Input, 8 Sensors, SNR = 100

*PRBS 1 has actuators 2 and 4 shifting phase with respect to actuators 1 and 3 every 40 steps (4.6 seconds); PRBS 2 switches phase every 80 steps (9.2 seconds). PRBS 2 is shown in Figure 5.3.6.

Mode	True Damped Frequency	Optimal Estimate SNR=30	Optimal Estimate SNR=100
1	1.1600	1.1771	1.1605
2	1.1800	1.2060	1.1801
3	1.2000	1.2279	1.2004
4	1.2200	1.2603	1.2198
5	1.2400	1.2524	1.2397
6	1.2600	1.2683	1.2601
7	1.2800	1.2960	1.2792
8	1.3000	1.3609	1.2997

(a) Identified Frequencies

Mode	Damping Ratio	Optimal Estimate SNR=30	Optimal Estimate SNR=100
1	.01	.0281	.0101
2	.01	.0167	.0105
3	.01	.0204	.0116
4	.01	.0257	.0122
5	.01	.1251	.0108
6	.01	.0524	.0105
7	.01	.0193	.0103
8	.01	.3447	.0100

(b) Identified Damping Ratios

Table 5.3.5: Identified Frequencies and Damping Ratios in the Presence of Modelling Errors, Optimal Input, 8 Sensors

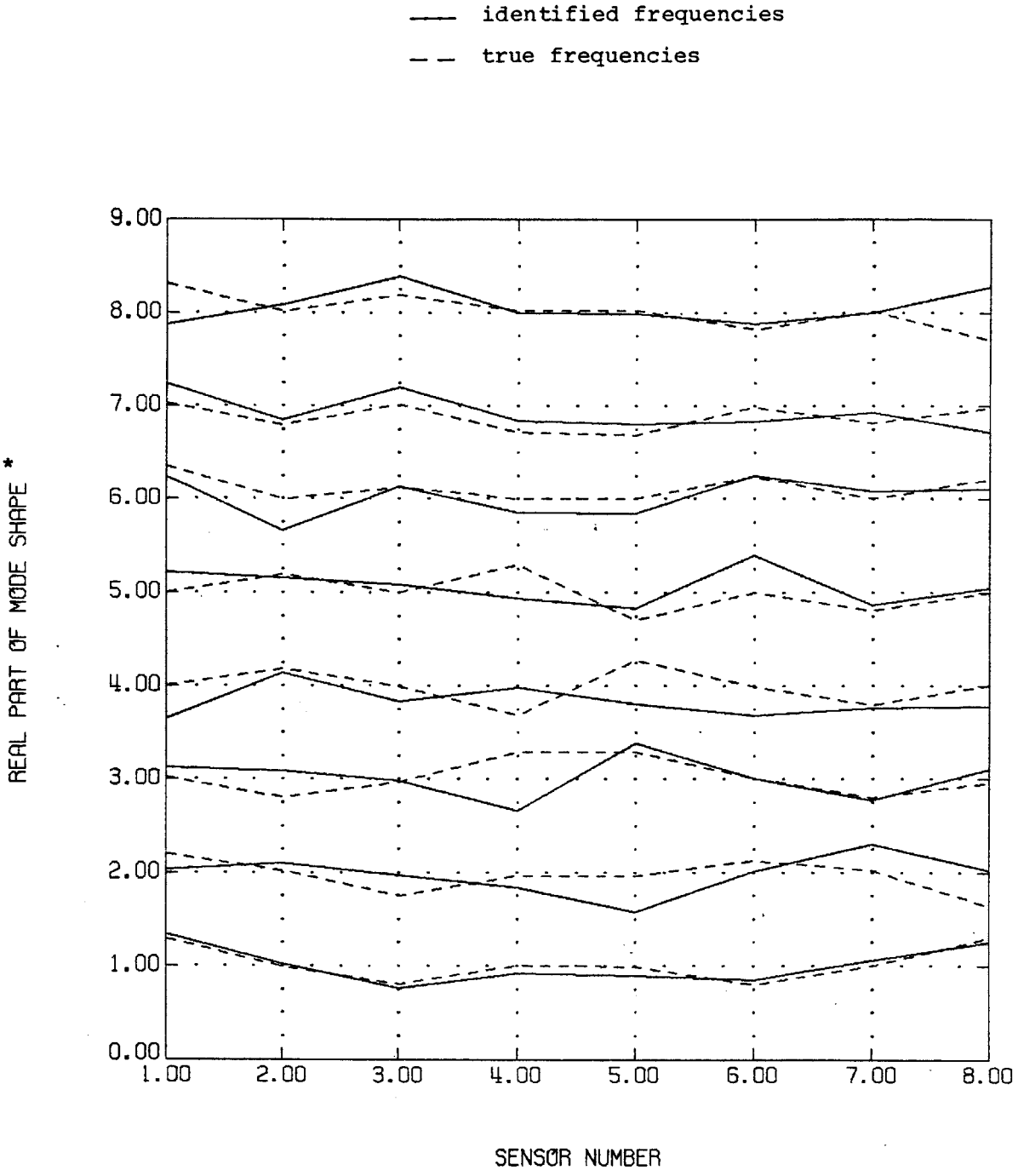


Figure 5.3.11: Real Part of Identified Mode Shapes,
8 Sensors, Modelling Errors, SNR = 30

*Integers are mode number; deflections about 0 for each integer are the mode shape

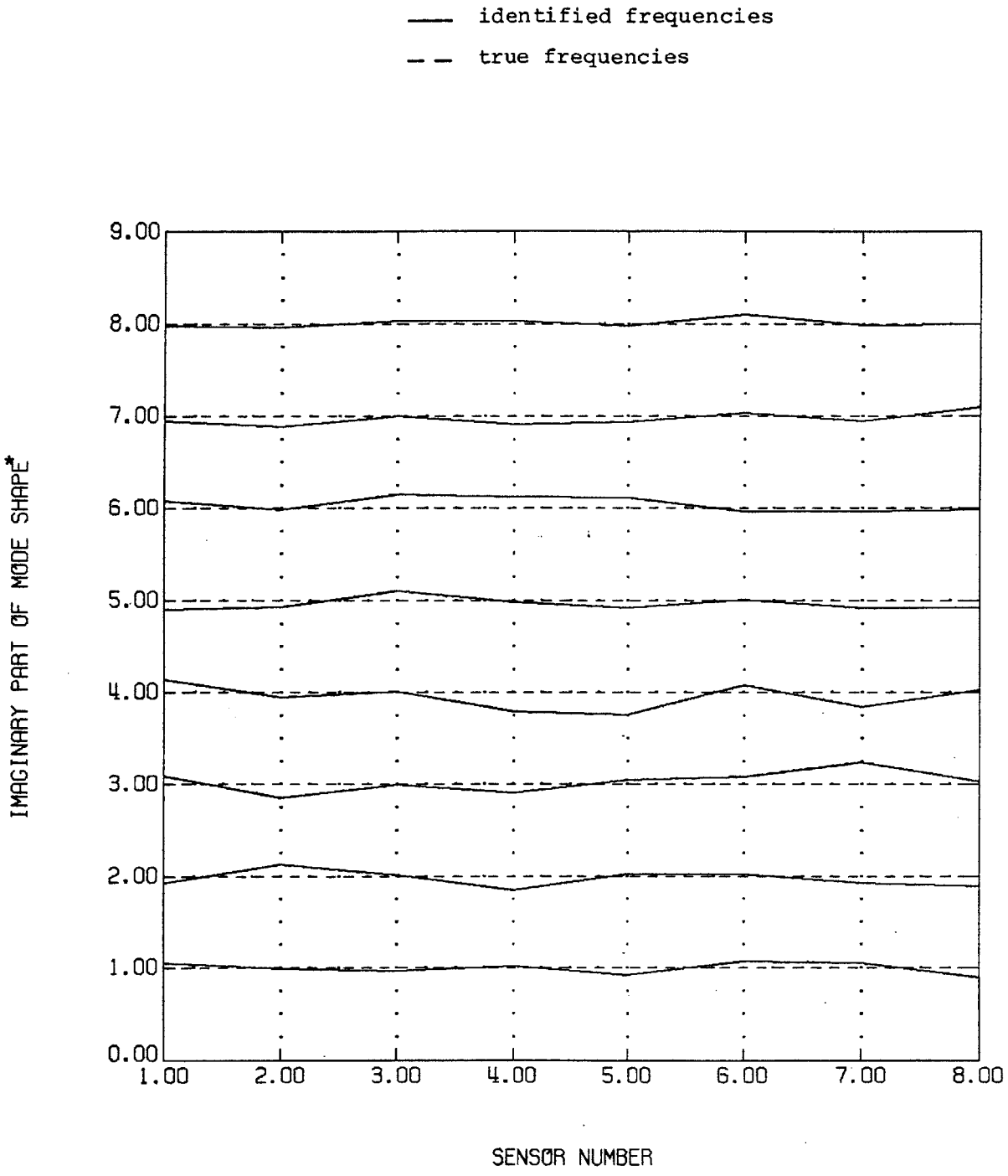


Figure 5.3.12: Imaginary Part of Identified Mode Shapes, 8 Sensors, Modelling Errors, SNR = 30

*Integers are mode number; deflections about 0 for each integer are the mode shape

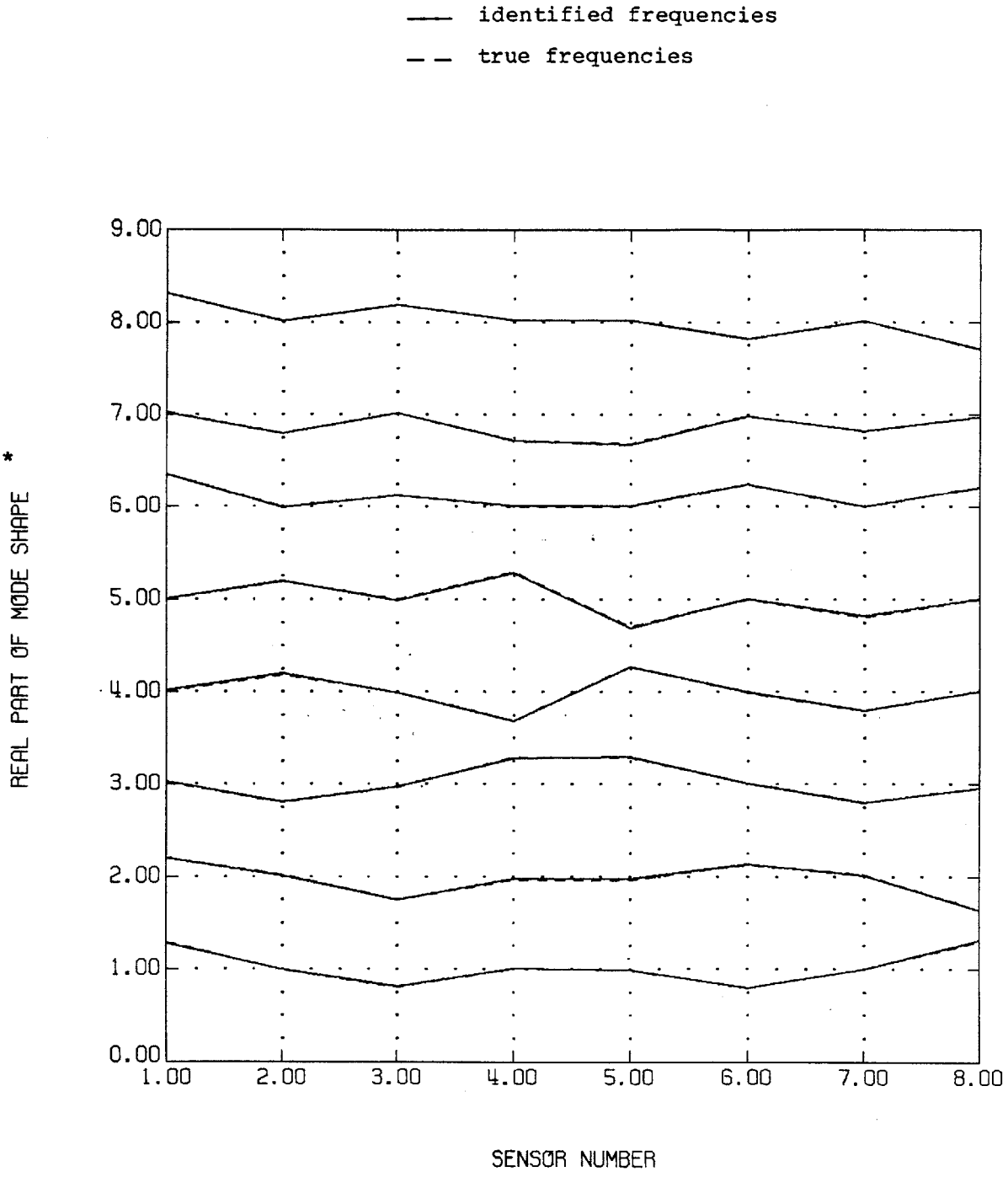


Figure 5.3.13: Real Part of Identified Mode Shapes,
8 Sensors, Modelling Errors, SNR = 100

*Integers are mode number; deflections about 0 for each integer are the mode shape

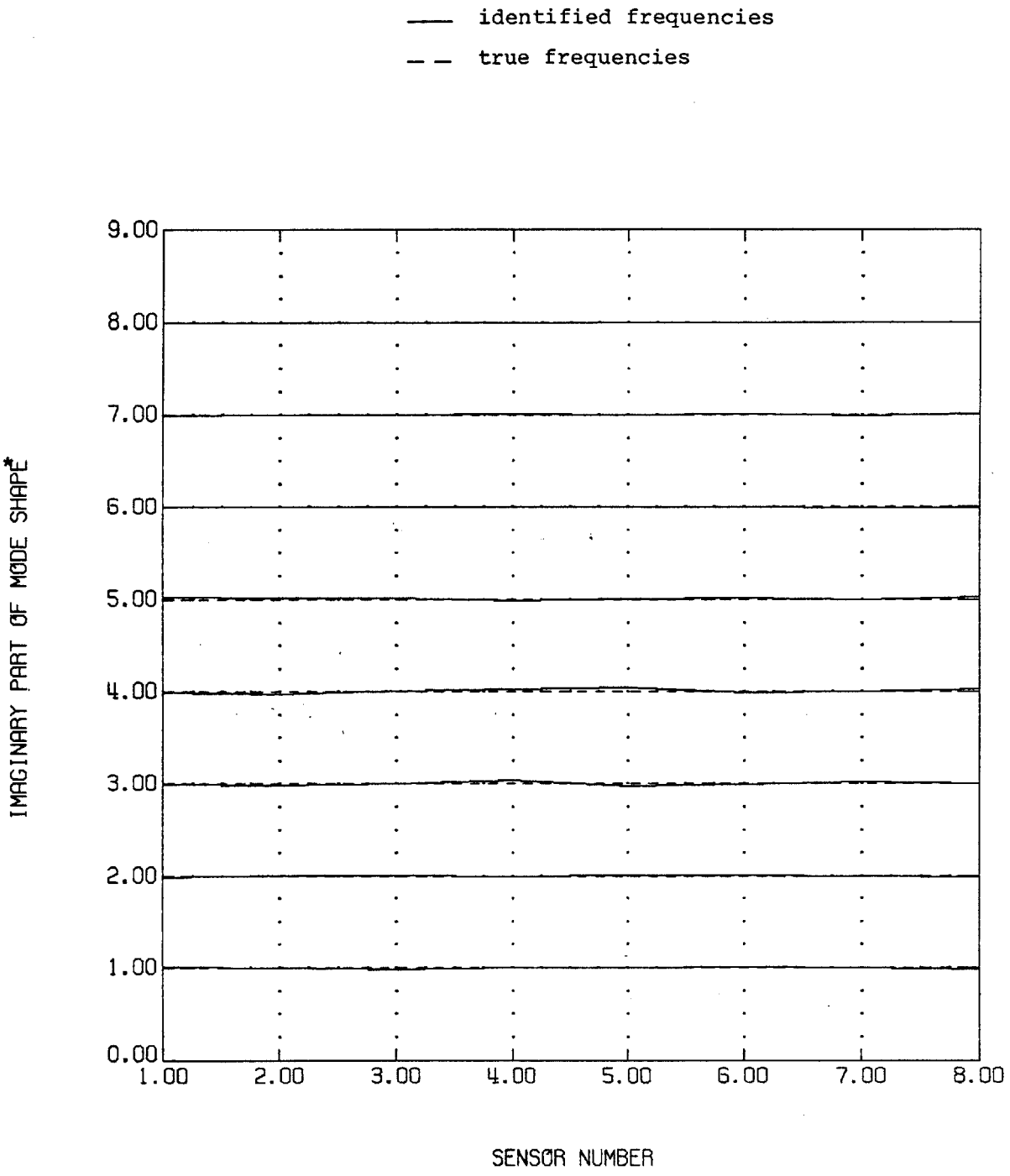


Figure 5.3.13: Imaginary Part of Identified Mode Shapes, 8 Sensors, Modelling Errors, SNR = 100

*Integers are mode number; deflections about 0 for each integer are the mode shape

Chapter 6: Conclusions and Recommended Future Work

The goal of this thesis was to create an identification package (model + inputs + algorithm) which was well-suited to the characteristics of real-time identification of LSSs.

The first step was to use the LSS characteristics (no prototype available in orbit, closely spaced modes, variations in apparent model size when modes are underexcited, need for a state space model) to select an algorithm. This established RLLS as the baseline algorithm. Simulation results on simple structures demonstrated that this algorithm:

(1) accurately identified LSSs,
(2) was as fast as predicted,
(3) was tolerant of variations in apparent numbers of modes,
(4) could identify closely spaced modes,
(5) generated a state space model which was useful for control and for verifying finite element models.

The next step was to examine ways to increase the cycle rate of the algorithm. First, reformulating the RLLS algorithm (as presented in the literature) as two interlocking Kalman filters eliminated some computations with no loss of identification accuracy. Second, two (related) ways were proposed which would decrease the cycle rate (inverse of CPU seconds per measurement $\geq$ sample rate for a real-time algorithm) and/or increase the number of modes with no increase in cycle rate. The better of these is parallel processing, which gives the increased cycle rate/more modes with no identification accuracy penalty. In the absence of parallel processing capability, the second approach uses the structure of the RLLS algorithm to simulate parallel processing i.e. do the blocks of computation serially. This approach was demonstrated in simulation to increase the cycle rate by a factor of four with little decrease in the accuracy of the identified parameters. In theory, parallel processing (and the suboptimal technique up to the limit of allowable identification error) eliminates

the cycle rate penalty for increasing model size. (Although, since total CPU time is constant, this is achieved at a similar penalty in length of data run.) In practice, the model size is still limited by finite accuracy and observability considerations.

The simulation results from the above work indicated that the standard PRBS input was not always adequate for good identification. An algorithm was developed for using the a priori knowledge of the system to generate an input which was more effective. The input is in the form of coefficients of sines, cosines, and exponentials rather than a table of values. This eliminates the storage requirements of a large table. Evaluating the cost function used to create the optimal input using the optimal input and PRBSs shows that even when there are small errors in the frequencies used to generate the optimal input, it still works better than a PRBS. Even when the frequencies are known very poorly, an input generated using the optimal input algorithm and assumed frequencies spaced evenly over an interval around the expected system frequency will outperform a PRBS.

Recommended Future Work

Finite Accuracy Errors

The space station example simulated a way to make a large system look smaller to handle closely spaced modes. It simulated using a bandpass filter to isolate clusters of modes. The problem is that it requires good knowledge of the frequencies of the gaps between clusters. Designing and evaluating a procedure to do this (e.g. by using an FFT to find the gaps, then a set of overlapping bandpass filters to identify the model) would be valuable.

Another possibility would be to explore ways to decrease the error or its effects (as has already been done to some extent by using the UDU^T formulation for the covariance matrices).

Optimal Input

The cost function used to create the optimal input was quadratic in the output sensitivities and input magnitude. The resulting input makes efficient use of the actuators by tailing off towards the end of the data run when a given amount of thrust does less good than earlier in the experiment. This is appropriate when fuel usage needs to be minimized. It would be interesting to compare these inputs to those generated using just the practical constraints that $|u| \leq u_{max}$ and $|x| \leq x_{deadband}$.

The optimal input as derived is a continuous input. In this thesis, the discrete input is obtained by applying sample-and-hold to this signal. When the sampling frequency is near the primary frequencies in the input, this can significantly distort the signal. A study of how significant this effect is might be informative.

Monitoring Apparent System Size

Since the RLLS variables $\underline{e}_n$ and $\underline{r}_n$ are a measure of the amount of new information contained in each additional order of the identifier, monitoring these might provide real-time estimates of the apparent system size. This could prevent spurious modes from being mistaken for real modes. Studies of a test (such as a whiteness test) to detect the presence of spurious modes could be useful.

Other Identification Issues

The space station results were generally poorer than the slinky results because of the close modal spacing. There is also the question of sensor and actuator placement. These areas need to be explored for best identifier performance.

Appendix A: Derivation of Recursive Lattice Least Squares

A.1 Derivation of RLLS as an Orthogonal Projection

The following derivation is reproduced from ref. 10, with some notational changes to match the rest of this thesis.

Suppose the system model is:

$$\underline{x}(t) = \sum_{i=1}^{N} G_i^{(N)}(t)\, \underline{x}(t-i)$$

where: $\underline{x}(t)$ = (m+p) × 1 vector of regressors = $[\underline{y}^T(t)\ \underline{u}^T(t)]^T$
$G_i^{(N)}$ = (m+p)×(m+p) matrix of regressor coefficients
N = assumed order of the model

Looking on this equation as a predictor of $\underline{x}(t)$, minimize in the least squares sense the prediction error:

$$\underline{e}_N(t) \triangleq \underline{x}(t) - \hat{\underline{x}}(t) = \underline{x}(t) - \sum_{i=1}^{N} G_i^{(N)}(t)\, \underline{x}(t-i) \qquad \text{(A1)}$$

The choice of $G_i^{(N)}$ which will give the minimum error is (ref. M, p. 19):

$$G^{(N)}(t) = \left[G_1^{(N)}(t) \cdots G_N^{(N)}(t)\right] = \underline{x}_t^T \underline{X}_{N,t}^T \left(X_{N,t} X_{N,t}^T\right)^{-1} \qquad \text{(A2)}$$

where: $X_{N,t} = \begin{bmatrix} \underline{x}(0) & \cdots & \underline{x}(t-1) \\ & & \vdots \\ \underline{0} & \underline{x}(0) \cdots & \underline{x}(t-N) \end{bmatrix}$ (an N(m+p)x(t+1) matrix) (A3)

$$\underline{x}_t = [\underline{x}(0) \cdots \underline{x}(t)]^T \qquad \text{(a (t+1)×(m+p) matrix)}$$

Substituting equation A2 into equation A1:

$$\underline{e}_N(t) = \underline{x}(t) - \underline{x}_t^T X_{N,t}^T (X_{N,t} X_{N,t}^T)^{-1} \text{ (last column of } X_{N,t})$$

$$= \underline{x}_t^T \underline{\Pi} - \underline{x}_t^T X_{N,t}^T (X_{N,t} X_{N,t}^T)^{-1} X_{N,t} \underline{\Pi}$$

$$\underline{e}_N(t) = \underline{x}_t^T (I - X_{N,t}^T (X_{N,t} X_{N,t}^T)^{-1} X_{N,t}) \underline{\Pi} = \underline{x}_t^T (I - P(X_{N,t})) \underline{\Pi} \qquad \text{(A4)}$$

where: $\underline{\Pi} \triangleq [0 \cdots 0\ 1]^T$ (a (t+1) column vector)

$P(X_{N,t}) = X_{N,t}^T (X_{N,t} X_{N,t}^T)^{-1} X_{N,t}$ = projection operator which projects on the space spanned by the rows of $X_{N,t}$

Similarly for the backward prediction errors:

$$\underline{r}_N(t-1) = \underline{x}(t-N-1) - \sum_{i=1}^{N} H_{N+1-i}^{(N)}(t)\, \underline{x}(t-i)$$

$$\underline{r}_N(t-1) = \underline{x}_t^{N+1^T} (I - P(X_{N,t})) \underline{\Pi} \qquad \text{(A5)}$$

where: $\underline{x}_t^{N+1} = [\underline{0} \cdots \underline{0}\ \underline{x}(0) \cdots \underline{x}(t-N-1)]^T = \underline{x}_t$ shifted N+1 times.

Note that the right hand sides of equation A4 and equation A5 are of the form $V^T[I-P(S)]W$ where V and W are matrices and vectors and P(S) is a projection operator, projecting on the space spanned by the rows of matrix S. Recursion formulas are derived by changing the

projection space. The space spanned by S+Y is the same as the space spanned by S+Y(I-P(S)), because (I-P(S)) is the complementary projection operator i.e.Y(I-P(S)) is orthogonal to S. Two matrices A and B are orthogonal if trace $(AB^T)=0$ (ref. X). In this case:

$$\text{trace}\left[Y(I-P(S))S^T\right] = \text{trace}\left[Y(I-S^T(SS^T)^{-1}S)S^T\right]$$
$$= \text{trace}\left[Y(S^T-S^T)\right] = 0$$

Since Y(I-P(s)) and S are orthogonal:

$$P(S+Y) = P(S+Y(I-P(S))) = P(S) + P(Y(I-P(S)))$$
$$= P(S) + (I-P(S))Y^T\left[Y(I-P(S))(I-P(S))Y^T\right]^{-1}Y(I-P(S))$$

Because I-P(S) is the complementary projection operator to P(S):

$$(I-P(S))P(S) = (I-S^T(SS^T)^{-1}S)S^T(SS^T)^{-1}S$$
$$= S^T(SS^T)^{-1}S - S^T(SS^T)^{-1}S=0$$

Substituting:

$$P(S+Y) = P(S) + (I-P(S))\ Y^T\left[Y(I-P(S))Y^T\right]^{-1}Y(I-P(S))$$

Or:

$$V^T(I-P(S+Y))W = V^T(I-P(S))W-V^T(I-P(S))Y^T\left[Y(I-P(S))Y^T\right]^{-1}Y(I-P(S))W \quad \text{(A6)}$$

The choice of Y in equation A6 determines whether the equation describes time, order, or both recursion:

I. To get order recursion, choose $Y = \underline{x}_t^{N+1^T}$:

$$X_{N,t} + Y = \begin{bmatrix} X_{N,t} \\ \underline{x}_t^{N+1\,T} \end{bmatrix} = X_{N+1,t} \qquad \text{(A7)}$$

II. To get time recursion, choose $Y=\underline{\Pi}^T$:

$$X_{N,t}+Y = \begin{bmatrix} \underline{x}(0) & \cdots & \underline{x}(t-1) \\ \underline{0} & & \vdots \\ & \underline{x}(0) \cdots \underline{x}(t-N) & \\ 0 & \cdots 0 \quad 1 & \end{bmatrix}$$

Using elementary row operations on the last row to eliminate the last column shows that this matrix spans the same space as:

$$\begin{bmatrix} \underline{x}(0) & \cdots & \underline{x}(t-2) & 0 \\ \underline{0} & & \vdots & \vdots \\ & \underline{x}(0) \cdots & \underline{x}(t-N-1) & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} & & & 0 \\ & X_{N,t-1} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

Since the upper matrix and bottom row are orthogonal:

$$P(X_{N,t}+Y) = P(\text{upper}) + P(\underline{\Pi}^T) = \begin{bmatrix} P(X_{N,t-1}) & \vdots \\ 0 \cdots & 0 \end{bmatrix} + \underline{\Pi}(\underline{\Pi}^T\underline{\Pi})^{-1}\,\underline{\Pi}^T$$

$$= \begin{bmatrix} P(X_{N,t-1}) & 0 \\ & \vdots \\ & 0 \\ 0 \cdots & 0 \quad 1 \end{bmatrix}$$

Or:

$$I - P(X_{N,t} + \underline{\pi}^T) = \begin{bmatrix} I - P(X_{N,t-1}) & 0 \\ & \vdots \\ 0 \quad \cdots & 0 \end{bmatrix} \tag{A8}$$

III. To get both time and order recursion, choose $Y = \underline{x}_t^T$:

$$X_{N,t} + Y = \begin{bmatrix} \underline{x}_t^T \\ X_{N,t} \end{bmatrix} = \begin{bmatrix} \underline{x}(0) & \cdots & \underline{x}(t) \\ \underline{0} \quad \ddots & & \vdots \\ & \underline{x}(0) \cdots & \underline{x}(t-N) \end{bmatrix} = X_{t+1,N+1} \begin{bmatrix} \underline{0}^T \\ I_t \end{bmatrix}$$

$$P(X_{N,t}+\underline{x}_t^T) = [\underline{0} \quad I_t] X_{t+1,N+1}^T \left(X_{t+1,N+1} \begin{bmatrix} \underline{0}^T \\ I_t \end{bmatrix} [\underline{0}^T \quad I_t] X_{t+1,N+1}^T\right)^{-1} X_{t+1,N+1} \begin{bmatrix} \underline{0}^T \\ I_x \end{bmatrix}$$

$$= [\underline{0} \quad I_t] X_{t+1,N+1}^T \left(X_{t+1,N+1} \; X_{t+1,N+1}^T\right)^{-1} X_{t+1,N+1} \begin{bmatrix} \underline{0}^T \\ I_t \end{bmatrix} \tag{A9}$$

When using this choice to get recursion formulas, note that:

$$\underline{x}_{t+1}^{N+2} = \begin{bmatrix} 0 \\ \underline{x}_t^{N+1} \end{bmatrix} \quad \text{or} \quad \underline{x}_{t+1}^{N+1^T} [\underline{0} \quad I_t] = \underline{x}_{t+1}^{N+2^T}$$

$$\underline{\pi}^T \underline{x}_t^{N+1} = \underline{x}(t-N-1) = \underline{\pi}^T [\underline{0} \quad I_t] \underline{x}_{t+1}^{N+2}$$

The next step is to use these projection rules to derive a recursive version of equation A4. Comparing equation A4 to equation A6 and using equations A7 and A5:

Y	$\underline{x}_t^{N+1^T}$
V	$\underline{x}_t$
W	$\underline{\pi}$
$V^T(I-P(S+Y))W$	$\underline{e}_{N+1}(t)$
$V^T(I-P(S))W$	$\underline{e}_N(t)$
$V^T(I-P(S))Y^T$	?1
$Y(I-P(S))Y^T$	?2
$Y(I-P(S))W$	$\underline{r}_N(t-1)$

Set: $?1 \triangleq F_N^T(t)$

$?2 \triangleq R_N(t-1)$

Three more recursive equations, for $\underline{r}_N$, F_N and R_N, are required to evaluate $\underline{e}_N$ recursively. Determining these:

Y	$\underline{x}_t^{N+1^T}$	$\underline{x}_t^T$	$\underline{\pi}^T$	$\underline{\pi}^T$
V	$\underline{x}_t$	$\underline{x}_t^{N+1}$	$\underline{x}_t^{N+1}$	$\underline{x}_t^{N+1}$
W	$\underline{\pi}$	$\underline{\pi}$	$\underline{x}_t$	$\underline{x}_t^{N+1}$
$V^T(I-P(S+Y))W$	$\underline{e}_{N+1}(t)$	$\underline{r}_{N+1}(t)$	$F_N(t-1)$	$R_N(t-1)$
$V^T(I-P(S))W$	$\underline{e}_N(t)$	$\underline{r}_N(t-1)$	$F_N(t)$	$R_N(t)$
$V^T(I-P(S))Y^T$	$F_N^T(t)$	$F_N(t)$	$\underline{r}_N(t-1)$	$\underline{r}_N(t-1)$
$Y(I-P(S))Y^T$	$R_N(t-1)$	?3	?4	?4
$Y(I-P(S))W$	$\underline{r}_N(t-1)$	$\underline{e}_N(t)$	$\underline{e}_N^T(t)$	$\underline{r}_N^T(t-1)$

Set: $?3 \triangleq R_N^*(t)$

$?4 \triangleq \beta_N(t)$

Two more equations are required:

Y	$\underline{x}_t^{N+1\,T}$	$\underline{x}_t^T$	$\underline{\pi}^T$	$\underline{\pi}^T$	$\underline{\pi}^T$	$\underline{x}_t^{N+1\,T}$
V	$\underline{x}_t$	$\underline{x}_t^{N+1}$	$\underline{x}_t^{N+1}$	$\underline{x}_t^{N+1}$	$\underline{x}_t$	$\underline{\pi}$
W	$\underline{\pi}$	$\underline{\pi}$	$\underline{x}_t$	$\underline{x}_t^{N+1}$	$\underline{x}_t$	$\underline{\pi}$
$V^T(I-P(S+Y))W$	$\underline{e}_{N+1}(t)$	$\underline{r}_{N+1}(t)$	$F_N(t-1)$	$R_N(t-2)$	$R_N^*(t-1)$	$\beta_{N+1}(t)$
$V^T(I-P(S))W$	$\underline{e}_N(t)$	$\underline{r}_N(t-1)$	$F_N(t)$	$R_N(t-1)$	$R_N^*(t)$	$\beta_N(t)$
$V^T(I-P(S))Y^T$	$F_N^T(t)$	$F_N(t)$	$\underline{r}_N(t-1)$	$\underline{r}_N(t-1)$	$\underline{e}_N(t)$	$\underline{r}_N^T(t-1)$
$Y(I-P(S))Y^T$	$R_N(t-1)$	$R_N^*(t)$	$\beta_N(t)$	$\beta_N(t)$	$\beta_N(t)$	$R_N(t-1)$
$Y(I-P(S))W$	$\underline{r}_N(t-1)$	$\underline{e}_N(t)$	$\underline{e}_N^T(t)$	$\underline{r}_N^T(t-1)$	$\underline{e}_N^T(t)$	$\underline{r}_N(t-1)$

Table A.1: Update Formulas for RLLS

The combinations $F_N^T(t)\ R_N^{-1}(t-1)$ and $F_N(t)\ R_N^{*-1}(t)$ are called reflection coefficients and are designated by K:

$$K_N(t) \triangleq F_N^T(t)\ R_N^{-1}(t-1)$$

$$K_N^*(t) \triangleq F_N(t)\ R_N^{*-1}(t)$$

The RLLS update equations (with the above substitutions and the help of equation A6) can be read from Table A.1:

$$\underline{e}_{N+1}(t) = \underline{e}_N(t) - K_N(t)\underline{r}_N(t-1)$$

$$\underline{r}_{N+1}(t) = \underline{r}_N(t-1) - K_N^*(t)\,\underline{e}_N(t)$$

$$F_N(t) = F_N(t-1) + \underline{r}_N(t-1)\,\underline{e}_N^T(t)/\beta_N(t)$$

$$R_N(t-1) = R_N(t-2) + \underline{r}_N(t-1)\underline{r}_N^T(t-1)/\beta_N(t)$$

$$R_N^*(t) = R_N^*(t-1) + \underline{e}_N(t)\,\underline{e}_N^T(t)/\beta_N(t)$$

$$\beta_{N+1}(t) = \beta_N(t) - \underline{r}_N^T(t-1)\,R_N^{-1}(t-1)\,\underline{r}_N(t-1)$$

These are the same equations as in Figure 2.4.1 except that the forgetting factor λ hasn't been included.

Recovering the coefficients $G_i^{(N)}$ of equation A1 requires finding a transformation between $K_N + K_N^*$ and the $G_i^{(N)}$. This relationship is derived in the following pages.

A.2 Derivation of RLLS as a Coordinate Transformation:

The following derivation is based on the one in Appendix 6.C in Ref. 20. This derivation differs from that in ref. 20 by emphasizing the change of coordinates aspect of the algorithm. This simplifies the algebra required in the ref. 20 derivation and makes the extension of the derivation (section A.3) trivial. Also, some notational changes were made to match the rest of this thesis and the parameter V was added (see note 1 under Extension of Derivation in section 2.4). Equation numbers with asterisks indicate equations which are listed in Figure 2.4.9.

The system model in ARMA form, as given in equation 2.4.1, is:

$$\hat{\underline{y}}(t) = \hat{\theta}(t)\,\underline{\phi}(t-1)$$

(assuming ν_i is the same for all i=1 to p, so that $\underline{\phi}_i(t)=\underline{\phi}(t)$). Writing the system order explicitly and including a prediction of the inputs as well as the outputs:

$$\tilde{\underline{x}}^{(n)}(t) = \hat{G}^{(n)}(t)\,\underline{\phi}^{(n)}(t-1) \qquad \text{(A10)}$$

where the first p rows of G equal θ. Since equation 2.4.3 will be used extensively in this derivation, it is repeated here for easy reference:

$$\underline{r}_n(t) = \underline{x}(t-n) - \hat{H}^{(n)}(t)\,\underline{\phi}^{(n)}(t) \tag{A11}$$

As described by equation A11, the $\underline{r}(t)$ are linear combinations of $\underline{\phi}(t)$ (where $\underline{\phi}^{(0)}(t) = \underline{0}$):

$$\underline{r}_o(t-1) = \underline{x}(t-1) = \underline{\phi}^{(1)}(t-1) = \begin{bmatrix} I_\alpha & 0 \end{bmatrix} \underline{\phi}^{(2)}(t-1)$$

$$\underline{r}_1(t-1) = \underline{x}(t-2) - \hat{H}^{(1)}(t-1)\,\underline{\phi}^{(1)}(t-1) = \begin{bmatrix} -\hat{H}^{(1)}(t-1) & I_\alpha \end{bmatrix} \phi^{(2)}(t-1)$$

$$\vdots$$

or: $$\tilde{\underline{\phi}}^{(n)}(t-1) \triangleq \begin{bmatrix} \underline{r}_o(t-1) \\ \vdots \\ \underline{r}_{n-1}(t-1) \end{bmatrix} = T^{(n)}(t)\,\underline{\phi}^{(n)}(t-1) \tag{A12}$$

where T is lower triangular with ones on the diagonal. Such a matrix is always invertible so:

$$\underline{\phi}^{(n)}(t-1) = T^{(n)^{-1}}(t)\,\tilde{\underline{\phi}}^{(n)}(t-1)$$

Define: $\begin{bmatrix} \hat{K}_o & \cdots & \hat{K}_{n-1} \end{bmatrix} = \tilde{G}^{(n)}(t) \triangleq \hat{G}^{(n)}(t)\,T^{(n)^{-1}}(t)$. Then equation A10 becomes:

$$\hat{\underline{x}}^{(n)}(t) = \tilde{G}^{(n)}(t)\,\tilde{\underline{\phi}}^{(n)}(t-1) \tag{A13}$$

$$= \hat{K}_o(t)\,\underline{r}_o(t-1) + \cdots + \hat{K}_{n-1}(t)\,\underline{r}_{n-1}(t-1) \tag{A14}$$

The covariance matrix P for the weighted least squares with forgetting factor solution of equation A10 is (ref. 20, p. 57):

$$P(t) = \left[\sum_{k=1}^{t} \lambda^{t-k} \underline{\phi}^{(n)}(k-1)\underline{\phi}^{(n)^T}(k-1)/V(k-1) \right]^{-1} \qquad \text{(A15)}$$

Likewise, for equation A13:

$$\tilde{P}(t) = \left[\sum_{k=1}^{t} \lambda^{t-k} \underline{\tilde{\phi}}^{(n)}(k-1)\underline{\tilde{\phi}}^{(n)^T}(k-1)/V(k-1) \right]^{-1}$$

$$= \left[\sum_{k=1}^{t} \lambda^{t-k} T^{(n)}(t) \underline{\phi}^{(n)}(k-1)\underline{\phi}^{(n)^T}(k-1) T^{(n)^T}(t)/V(k-1) \right]^{-1}$$

$$= T^{(n)^{-T}}(t) \left[\sum_{k=1}^{t} \lambda^{t-k} \underline{\phi}^{(n)}(k-1)\underline{\phi}^{(n)^T}(k-1)/V(k-1) \right]^{-1} T^{(n)^{-1}}(t)$$

$$= T^{(n)^{-T}}(t) P(t) T^{(n)^{-1}}(t)$$

or: $$P(t) = T^{(n)^T}(t) \tilde{P}(t) T^{(n)}(t) \qquad \text{(A16)}$$

The $\underline{r}(t)$ were specifically chosen so that $E\{\underline{r}_i(t) \underline{r}_j^T(t)\}=0$ for $i \neq j$, so that $\tilde{P}(t)$ is asymptotically block diagonal with blocks of dimension $\alpha \times \alpha$. (Note that if α were 1, then $\tilde{P}(t)$ asymptotically equals D in the UDU^T decomposition of $P(t)$ and $T^{(n)^T}$ asymptotically equals U).

Relationship Between Transformed and Untransformed Coefficients:

Comparing equation A14 and A10 order by order gives the relationship between $G^{(n)}$ and $\tilde{G}^{(n)}$ i.e. $G^{(n)}$ and K_n:

Order 1: $$\hat{\underline{x}}^{(1)}(t) = \hat{K}_o(t) \underline{r}_o(t-1) = \hat{G}^{(1)}(t) \underline{x}(t-1)$$

From equation A11, $\underline{r}_o(t-1) = \underline{x}(t-1)$:

$$\hat{K}_o(t)\ \underline{x}(t-1) = \hat{G}^{(1)}(t)\ \underline{x}(t-1) \text{ or } \hat{K}_o(t) = \hat{G}^{(1)}(t) \tag{A17}$$

nth order:

$$\underline{\hat{x}}^{(n+1)}(t) = \underline{\hat{x}}^{(n)}(t) + \hat{K}_n(t)\ \underline{r}_n(t-1)$$

$$= \hat{G}^{(n)}(t)\ \underline{\phi}^{(n)}(t-1) + \hat{K}_n(t)\ \underline{r}_n(t-1) = \hat{G}^{(n+1)}(t)\ \underline{\phi}^{(n+1)}(t-1) \tag{A18}$$

From equation A11, $\underline{r}_n(t-1) = \underline{x}(t-n-1) - \hat{H}^{(n)}(t-1)\underline{\phi}^{(n)}(t-1)$

$$= \left[-H^{(n)}(t-1) \quad I_\alpha\right]\ \underline{\phi}^{(n+1)}(t-1): \tag{A19}$$

$$\hat{G}^{(n+1)}(t)\underline{\phi}^{(n+1)}(t-1) = \hat{G}^{(n)}(t)\underline{\phi}^{(n)}(t-1) + \hat{K}_n(t)\left[-\hat{H}^{(n)}(t-1) \quad I_\alpha\right]\underline{\phi}^{(n+1)}(t-1)$$

$$= \{ [\hat{G}^{(n)}(t) \quad 0] + \hat{K}_n(t) [-\hat{H}^{(n)}(t-1) \quad I_\alpha] \} \underline{\phi}^{(n+1)}(t-1)$$

or: $$\hat{G}^{(n+1)}(t) = [\hat{G}^{(n)}(t) \quad 0] + \hat{K}_n(t) [-\hat{H}^{(n)}(t-1) \quad I_\alpha] \qquad *(A20)$$

with: $$\hat{G}^{(1)}(t) = \hat{K}_o(t)$$

Using equation A20 requires a way to generate $H^{(n)}(t)$. A recursive equation for $H^{(n)}(t)$ can be derived by considering the counterpart of equation A11 which is obtained from equation A10. The $\underline{r}_n$ are backward prediction errors (or residuals), where $\underline{r}_n(t)$ is that part of $\underline{x}(t-n)$ which can't be predicted from $\underline{x}(t), \cdots, \underline{x}(t-n+1)$. Equation A10 can be used to generate forward prediction errors $\underline{e}_n(t)$:

$$\underline{e}_n(t) = \underline{x}(t) - \hat{G}^{(n)}(t) \, \underline{\phi}^{(n)}(t-1) \qquad (A21)$$

This leads to alternate expressions for $\hat{\underline{x}}(t)$ which are similar to equations A10 and A14. From equation A11:

$$\hat{\underline{x}}^{(n)}(t-n) = \hat{H}^{(n)}(t) \, \underline{\phi}^{(n)}(t) \qquad (A22)$$

$$= \hat{K}_o^*(t-n+1) \, \underline{e}_o(t-n+1) + \cdots + \hat{K}_{n-1}^*(t) \, \underline{e}_{n-1}(t) \qquad (A23)$$

The time indexing in equation A23 can be understood by a change of notation. Define $t' \triangleq t-n$. Then equation A23 becomes:

$$\hat{\underline{x}}^{(n)}(t') = \hat{K}_o^*(t'+1) \, \underline{e}_o(t'+1) + \cdots + \hat{K}_{n-1}^*(t'+n) \, \underline{e}_{n-1}(t'+n)$$

Thus, the low order terms do not depend on the number of terms in the equation, as must be true if the $\underline{e}_n$ are orthogonal. Another way to look at it is that the ith term is a linear combination of $\underline{x}(t'+1), \cdots, \underline{x}(t'+i)$ just as in equation A14 the ith term is a linear combination of $\underline{x}(t-1), \cdots, \underline{x}(t-i)$.

Deriving the recursion for $H^{(n)}$ by comparing equations A22 and A23 order by order:

1st order:

$$\underline{\hat{x}}^{(1)}(t) = \hat{H}^{(1)}(t+1)\,\underline{\phi}^{(1)}(t+1) = \hat{K}_o^*(t+1)\underline{e}_o(t+1) = \hat{H}^{(1)}(t+1)\underline{x}(t+1)$$

From equation A21, $\underline{e}_o(t+1) = \underline{x}(t+1)$:

$$\hat{H}^{(1)}(t) = \hat{K}_o^*(t) \tag{A24}$$

nth order:

$$\begin{aligned}\underline{\hat{x}}^{(n)}(t) &= \hat{H}^{(n)}(t+n)\,\underline{\phi}^{(n)}(t+n)\\ &= \hat{K}_o^*(t+1)\,\underline{e}_o(t+1)+\cdots+\hat{K}_{n-2}^*(t+n-1)\underline{e}_{n-2}^*(t+n-1)+\hat{K}_{n-1}^*(t+n)\underline{e}_{n-1}(t+n)\\ &= \underline{\hat{x}}^{(n-1)}(t) + \hat{K}_{n-1}^*(t+n)\underline{e}_{n-1}(t+n)\end{aligned}$$

$$\hat{H}^{(n)}(t+n)\underline{\phi}^{(n)}(t+n) = \hat{H}^{(n-1)}(t+n-1)\underline{\phi}^{(n)}(t+n-1)+\hat{K}_{n-1}^*(t+n)\underline{e}_{n-1}(t+n) \tag{A25}$$

Shifting the time down by n and substituting from equation A21:

$$\begin{aligned}\hat{H}^{(n)}(t)\underline{\phi}^{(n)}(t) &= \hat{H}^{(n-1)}(t-1)\underline{\phi}^{(n)}(t-1)+\hat{K}_{n-1}^*(t)\left[\underline{x}(t)-\hat{G}^{(n-1)}(t)\underline{\phi}^{(n-1)}(t-1)\right]\\ &= \left\{\left[0 \quad \hat{H}^{(n-1)}(t-1)\right]+\hat{K}_{n-1}^*(t)\left[I_\alpha \quad -\hat{G}^{(n-1)}(t)\right]\right\}\underline{\phi}^{(n)}(t)\end{aligned}$$

or: $$\hat{H}^{(n)}(t) = \left[0 \quad \hat{H}^{(n-1)}(t-1)\right] + \hat{K}_{n-1}^*(t)\left[I_\alpha \quad -\hat{G}^{(n-1)}(t)\right] \tag{A26}$$

with: $\hat{H}^{(1)}(t) = \hat{K}_o^*(t)$

Generating r_n and e_n:

The next step is to derive equations for generating $\underline{r}_n$ and $\underline{e}_n$. From equation A11:

$$\underline{x}(t-n)=\underline{r}_n(t)+\hat{H}^{(n)}(t)\underline{\phi}^{(n)}(t)=\underline{r}_{n-1}(t-1)+\hat{H}^{(n-1)}(t-1)\underline{\phi}^{(n-1)}(t-1)$$

or: $$\underline{r}_n(t)=\underline{r}_{n-1}(t-1)-\hat{H}^{(n)}(t)\underline{\phi}^{(n)}(t)+\hat{H}^{(n-1)}(t-1)\underline{\phi}^{(n-1)}(t-1)$$

Substituting from equation A25 (after shifing the time in A25 down by n):

$$\underline{r}_n(t) = \underline{r}_{n-1}(t-1) - \hat{K}^{*}_{n-1}(t)\ \underline{e}_{n-1}(t) \qquad *(A27)$$

Similarly, from equation A21:

$$\underline{x}(t) = \underline{e}_n(t) + \hat{G}^{(n)}(t)\ \underline{\phi}^{(n)}(t-1) = \underline{e}_{n-1}(t) + \hat{G}^{(n-1)}(t)\ \underline{\phi}^{(n-1)}(t-1)$$

or: $$\underline{e}_n(t) = \underline{e}_{n-1}(t) - \hat{G}^{(n)}(t)\ \underline{\phi}^{(n)}(t-1) + \hat{G}^{(n-1)}(t)\ \underline{\phi}^{(n-1)}(t-1)$$

Substituting from equation A18 (after shifting A18 down one order):

$$\underline{e}_n(t) = \underline{e}_{n-1}(t) - \hat{K}_{n-1}(t)\ \underline{r}_{n-1}(t-1) \qquad *(A28)$$

Generating K_n and K_n^*:

From ref. N, p. 57, the weighted least squares with forgetting factor solutions for K_n (from equation A28) and K_n^* (from equation A27) are:

$$\hat{K}_n(t) = \left\{ \sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \lambda(j) \right] \underline{e}_n(k)\underline{r}_n^T(k-1)\beta_n^{-1}(k) \right\} x$$

$$\left\{ \sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \lambda(j) \right] \underline{r}_n(k-1)\underline{r}_n^T(k-1)\beta_n^{-1}(k) \right\}^{-1}$$

$$\hat{K}_n(t) \triangleq F_n^T(t) R_n^{-1}(t-1) \qquad *(A29)$$

$$\hat{K}_n^*(t) = \left(\sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \lambda(j) \right] \underline{r}_n(k-1) \underline{e}_n^T(k) \beta_n^{-1}(k) \right) \left(\sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \lambda(j) \right] \underline{e}_n(k) \underline{e}_n^T(k) \beta_n^{-1}(k) \right)^{-1}$$

$$\hat{K}_n^*(t) \triangleq F_n(t)\, R_n^{*-1}(t) \qquad *(A30)$$

where: $\beta_n(k)$ = as yet undetermined weighting factor dependent on the measurement noise variance V(k) and the order n. There is a suboptimal version of this algorithm called a gradient lattice which sets this parameter to 1.

$$F_n(t) \triangleq \sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \lambda(j) \right] \underline{r}_n(k-1) \underline{e}_n^T(k) \beta_n^{-1}(k) \qquad (A31)$$

$$R_n(t-1) \triangleq \sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \lambda(j) \right] \underline{r}_n(k-1) \underline{r}_n^T(k-1) \beta_n^{-1}(k) \qquad (A32)$$

$$R_n^*(t) \triangleq \sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \lambda(j) \right] \underline{e}_n(k) \underline{e}_n^T(k) \beta_n^{-1}(k) \qquad (A33)$$

Deriving these in recursive form:

$$F_n(t) = \sum_{k=1}^{t-1} \left[\prod_{j=k+1}^{t} \lambda(j) \right] \underline{r}_n(k-1) \underline{e}_n^T(k) \beta_n^{-1}(k) + \underline{r}_n(t-1) \underline{e}_n^T(t) / \beta_n(t)$$

$$= \lambda(t)\, F_n(t-1) + \underline{r}_n(t-1)\, \underline{e}_n^T(t) / \beta_n(t) \qquad *(A34)$$

Similarly for equations A32 and A33:

$$R_n(t-1) = \lambda(t)\, R_n(t-2) + \underline{r}_n(t-1)\, \underline{r}_n^T(t-1) / \beta_n(t) \qquad *(A35)$$

$$R_n^*(t) = \lambda(t)\, R_n^*(t-1) + \underline{e}_n(t)\, \underline{e}_n^T(t) / \beta_n(t) \qquad *(A36)$$

Generating $\beta_n(t)$:

The β_n can be calculated by transforming the weighting factors from the LS form. The LS solution is (ref. N, p. 57):

$$\underline{\hat{x}}^{(n)}(t) = \hat{G}^{(n)}(t)\,\underline{\phi}^{(n)}(t-1) \tag{A10}$$

$$\hat{G}^{(n)}(t) = \hat{G}^{(n)}(t-1) + [\underline{x}(t) - \hat{G}^{(n)}(t-1)\underline{\phi}^{(n)}(t-1)]\underline{k}^{(n)^T}(t) \tag{A37}$$

$$R^{(n)}(t-1) = \sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \lambda(j)\right] \underline{\phi}^{(n)}(k-1)\,\underline{\phi}^{(n)^T}(k-1)/V(k) \tag{A38}$$

$$\underline{k}^{(n)}(t) = R^{(n)^{-1}}(t-1)\,\underline{\phi}^{(n)}(t-1)/V(t) \tag{A39}$$

The time indexing on $R^{(n)}(t)$ has been changed from the standard version. In RLLS, the regressors used to calculate $\underline{\hat{x}}^{(n)}(t)$ are available at time t-1 (e.g. $\underline{u}(t)$ is not included). The noise statistics and forgetting factor (see equation A38) are needed to time t, so that $R^{(n)}(t-1)$ apparently depends on data at time t. However, the noise statistics and forgetting factor are constant for RLLS (see Approximation Used in Equation A53, at the end of section A.2), so the notation $R^{(n)}(t-1)$ is appropriate. As indicated by equation A39, $R^{(n)}(t-1)$ isn't used until time t, but it can be calculated at time t-1. From equation A38, A12, and A16 (with the new notation, P(t) is written as P(t-1) and equals $R^{(n)^{-1}}(t-1)$):

$$\underline{k}^{(n)^T}(t)\,\underline{\phi}^{(n)}(t-1) = \underline{\phi}^{(n)^T}(t-1)\,R^{(n)^{-1}}(t-1)\,\underline{\phi}^{(n)}(t-1)/V(t)$$

$$= \underline{\tilde{\phi}}^{(n)^T}(t-1)T^{(n)^{-T}}(t)\,T^{(n)^T}(t)\tilde{P}(t)T^{(n)}(t)T^{(n)^{-1}}(t)\underline{\tilde{\phi}}^{(n)}(t-1)/V(t)$$

$$= \underline{\tilde{\phi}}^{(n)^T}(t-1)\tilde{P}(t)\,\underline{\tilde{\phi}}^{(n)}(t-1)/V(t)$$

From equation A12 and since the $\underline{r}_i$ are uncorrelated:

$$V(t)\underline{k}^{(n)^T}(t)\,\underline{\phi}^{(n)}(t-1) = \left[\underline{r}_o^T(t-1) \cdots \underline{r}_{n-1}^T(t-1)\right]\begin{bmatrix} R_o^{-1}(t-1) & & 0 \\ & \ddots & \\ 0 & & R_{n-1}^{-1}(t-1) \end{bmatrix}\begin{bmatrix} \underline{r}_o(t-1) \\ \vdots \\ \underline{r}_{n-1}(t-1) \end{bmatrix}$$

$$\underline{k}^{(n)^T}(t)\,\underline{\phi}^{(n)}(t-1) = \sum_{i=0}^{n-1} \underline{r}_i^T(t-1)R_o^{-1}(t-1)\underline{r}_i(t-1)/V(t) \qquad \text{(A40)}$$

or:

$$\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1) = \underline{k}^{(n-1)^T}(t)\underline{\phi}^{(n-1)}(t-1)+\underline{r}_{n-1}^T(t-1)R_{n-1}^{-1}(t-1)\underline{r}_{n-1}(t-1)/V(t)$$

Note the similarity of the term in brackets in equation A37 to equation A21. Call this $\bar{\underline{e}}_n(t)$:

$$\bar{\underline{e}}_n(t) \triangleq \underline{x}(t) - \hat{G}^{(n)}(t-1)\,\underline{\phi}^{(n)}(t-1) \qquad \text{(A42)}$$

Similarly, from equation A11:

$$\bar{\underline{r}}_n(t) \triangleq \underline{x}(t-n) - \hat{H}^{(n)}(t-1)\,\underline{\phi}^{(n)}(t) \qquad \text{(A43)}$$

If the derivations for equations A27 and A28 are repeated using these estimators:

$$\bar{\underline{r}}_n(t) = \bar{\underline{r}}_{n-1}(t-1) - \hat{K}^*_{n-1}(t-1)\,\bar{\underline{e}}_{n-1}(t) \qquad \text{(A44)}$$

$$\bar{\underline{e}}_n(t) = \bar{\underline{e}}_{n-1}(t) - \hat{K}_{n-1}(t-1)\bar{\underline{r}}_{n-1}(t-1) \qquad \text{(A45)}$$

Substituting equation A42 into equation A37:

$$\hat{G}^{(n)}(t) = \hat{G}^{(n)}(t-1) + \underline{\bar{e}}_n(t)\underline{k}^{(n)^T}(t) \tag{A46}$$

To relate $\underline{e}_n(t)$ and $\underline{\bar{e}}_n(t)$, combine equations A21, A46, and A42:

$$\begin{aligned}
\underline{e}_n(t) &= \underline{x}(t) - \hat{G}^{(n)}(t)\,\underline{\phi}^{(n)}(t-1) \\
&= \underline{x}(t) - \hat{G}^{(n)}(t-1)\underline{\phi}^{(n)}(t-1) - \underline{\bar{e}}_n(t)\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1) \\
&= \underline{\bar{e}}_n(t) - \underline{\bar{e}}_n(t)\,\underline{k}^{(n)}(t)\,\underline{\phi}^{(n)}(t-1) \\
&= \left(1 - \underline{k}^{(n)^T}(t)\,\underline{\phi}^{(n)}(t-1)\right)\underline{\bar{e}}_n(t)
\end{aligned} \tag{A47}$$

To relate $\underline{r}_n$ and $\underline{\bar{r}}_n$, the RLS solution to equation A22 is needed:

$$\underline{\hat{x}}^{(n)}(t-n) = \hat{H}^{(n)}(t)\,\underline{\phi}^{(n)}(t) \tag{A22}$$

$$\hat{H}^{(n)}(t) = \hat{H}^{(n)}(t-1)\left[\underline{x}(t-n) - \hat{H}^{(n)}(t-1)\,\underline{\phi}^{(n)}(t)\right]\underline{l}^{(n)^T}(t) \tag{A48}$$

$$Q^{(n)}(t) = \sum_{k=1}^{t}\left[\prod_{j=k+1}^{t}\lambda(j)\right]\underline{\phi}^{(n)}(k)\,\underline{\phi}^{(n)^T}(k)/V(k) \tag{A49}$$

$$\underline{l}^{(n)}(t) = Q^{(n)^{-1}}(t)\,\underline{\phi}^{(n)}(t)/V(t) \tag{A50}$$

Substituting equation A43 into A48:

$$\hat{H}^{(n)}(t) = \hat{H}^{(n)}(t-1) + \underline{\bar{r}}_n(t)\,\underline{l}^{(n)^T}(t) \tag{A51}$$

Combining equations A11, A51, and A43:

$$\underline{r}_n(t) = \underline{x}(t-n) - \hat{H}^{(n)}(t)\ \underline{\phi}^{(n)}(t)$$

$$= \underline{x}(t-n) - \hat{H}^{(n)}(t-1)\ \underline{\phi}^{(n)}(t) - \bar{\underline{r}}_n(t)\ \underline{l}^{(n)^T}(t)\ \underline{\phi}^{(n)}(t)$$

$$= \bar{\underline{r}}_n(t) - \bar{\underline{r}}_n(t)\ \underline{l}^{(n)^T}(t)\ \underline{\phi}^{(n)}(t)$$

$$= \left(1 - \underline{l}^{(n)^T}(t)\ \underline{\phi}^{(n)}(t)\right) \bar{\underline{r}}_n(t) \qquad \text{(A52)}$$

Comparing equations A49 to A38 and A50 to A39 shows that, for measurement noise statistics and forgetting factor constant:

$$\underline{l}^{(n)}(t) = \underline{k}^{(n)}(t+1) \qquad \text{(A53)}$$

(For a detailed discussion of this equivalence, see "Approximation Used in Equation A53" at the end of section A.2.) Since the noise statistics and forgetting factor must be constant, use V in place of V(t) and λ in place of $\lambda(t)$ in the rest of this derivation. Substituting equation A53 to A52:

$$\underline{r}_n(t) = \left(1 - \underline{k}^{(n)^T}(t+1)\ \underline{\phi}^{(n)}(t)\right) \bar{\underline{r}}_n(t) \qquad \text{(A54)}$$

To compare the estimates of K_n and K_n^* from equations A29 and A30 to those for $G^{(n)}$ and $H^{(n)}$ in equations A46 and A51, the K_n and K_n^* equations have to be converted to the same form as the $G^{(n)}$ and $H^{(n)}$ equations. Starting with equation A29, substituting A34, A29, A35, A47, A54, and A45:

$$\hat{K}_n(t) = F_n^T(t)\ R_n^{-1}\ (t-1)$$

$$= \left(\lambda\ F_n^T(t-1) + \underline{e}_n(t)\ \underline{r}_n^T(t-1)/\beta_n(t)\right) R_n^{-1}(t-1)$$

$$= \left[\lambda\, \hat{K}_n(t-1)\, R_n(t-2) + \underline{e}_n(t)\, \underline{r}_n^T(t-1)/\beta_n(t)\right] R_n^{-1}(t-1)$$

$$= \left(\hat{K}_n(t-1)\left[R_n(t-1) - \underline{r}_n(t-1)\, \underline{r}_n^T(t-1)/\beta_n(t)\right] + \underline{e}_n(t)\, \underline{r}_n^T(t-1)/\beta_n(t)\right) R_n^{-1}(t-1)$$

$$= \hat{K}_n(t-1) - \left\{\hat{K}_n(t-1)\, \underline{r}_n(t-1)\, \underline{r}_n^T(t-1)/\beta_n(t) + \underline{e}_n(t)\underline{r}_n^T(t-1)/\beta_n(t)\right\} R_n^{-1}(t-1)$$

$$= \hat{K}_n(t-1) + \left(\underline{e}_n(t) - \hat{K}_n(t-1)\, \underline{r}_n(t-1)\right)\underline{r}_n^T(t-1)\, R_n^{-1}(t-1)/\beta_n(t)$$

$$= \hat{K}_n(t-1) + \left(\underline{\bar{e}}_n(t) - \hat{K}_n(t-1)\underline{\bar{r}}_n(t-1)\right)\underline{r}_n^T(t-1) R_n^{-1}(t-1)\left(1 - \underline{k}^{(n)T}(t)\underline{\phi}^{(n)}(t-1)\right)/\beta_n(t)$$

$$\hat{K}_n(t) = \hat{K}_n(t-1) + \underline{\bar{e}}_{n+1}(t)\underline{r}_n^T(t-1) R_n^{-1}(t-1)\left(1 - \underline{k}^{(n)T}(t)\underline{\phi}^{(n)}(t-1)\right)/\beta_n(t) \tag{A55}$$

Similarly, using equations A30, A34, A30, A36, A47, A54, and A44:

$$\hat{K}_n^*(t) = \hat{K}_n^*(t-1) + \underline{\bar{r}}_{n+1}(t)\underline{e}_n^T(t) R_n^{*-1}(t)\left(1 - \underline{k}^{(n)T}(t)\, \underline{\phi}^{(n)}(t-1)\right)/\beta_n(t)$$

To compare $\hat{G}^{(n)}$ and $\hat{K}_{n-1}$, start with equation A46 then substitute A20:

$$\hat{G}^{(n+1)}(t) = \hat{G}^{(n+1)}(t-1) + \underline{\bar{e}}_{n+1}(t)\, \underline{k}^{(n+1)T}(t)$$

$$\left[\hat{G}^{(n)}(t)\ 0\right] + \hat{K}_n(t)\left[-\hat{H}^{(n)}(t-1)\ I\right] = \left[\hat{G}^{(n)}(t-1)\ 0\right] + \hat{K}_n(t-1)\left[-\hat{H}^{(n)}(t-2)\ I\right] + \underline{\bar{e}}_{n+1}(t)\underline{k}^{(n+1)T}(t)$$

$$\left[\hat{G}^{(n)}(t) - \hat{G}^{(n)}(t-1)\quad 0\right] + \hat{K}_n(t)\left[-\hat{H}^{(n)}(t-1)\ I\right] - \hat{K}_n(t-1)\left[-\hat{H}^{(n)}(t-2)\ I\right] = \underline{\bar{e}}_{n+1}(t)\, \underline{k}^{(n+1)T}(t)$$

Multiply from the right by $\underline{\phi}^{(n+1)}(t-1) = [\underline{\phi}^{(n)^T}(t-1) \quad \underline{x}^T(t-n)]^T$:

$$[\hat{G}^{(n)}(t)-\hat{G}^{(n)}(t-1)]\underline{\phi}^{(n)}(t-1)+\hat{K}_n(t)[-\hat{H}^{(n)}(t-1)\ I]\underline{\phi}^{(n+1)}(t-1)-$$

$$\hat{K}_n(t-1)[-\hat{H}^{(n)}(t-2)\ I]\underline{\phi}^{(n+1)}(t-1) = \bar{\underline{e}}_{n+1}(t)\underline{k}^{(n+1)^T}(t)\ \underline{\phi}^{(n+1)}(t-1)$$

Substituting from equations A46, A19, and the equivalent of A19 for $\bar{\underline{r}}_n$ (see equation A43):

$$\bar{\underline{e}}_n(t)\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)+\hat{K}_n(t)\underline{r}_n(t-1)-\hat{K}_n(t-1)\bar{\underline{r}}_n(t-1)=$$

$$\bar{\underline{e}}_{n+1}(t)\underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1)$$

Substituting from equation A55:

$$\bar{\underline{e}}_n(t)\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)+\hat{K}_n(t)\underline{r}_n(t-1)-\hat{K}_n(t-1)\bar{\underline{r}}_n(t-1)\ +$$

$$\bar{\underline{e}}_{n+1}(t)\underline{r}_n^T(t-1)R_n^{-1}(t-1)\bar{\underline{r}}_n(t-1)[1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)]/\beta_n(t)=$$

$$\bar{\underline{e}}_{n+1}(t)\underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1)$$

Substituting from equation A54:

$$\bar{\underline{e}}_n(t)\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)+(1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1))\hat{K}_n(t)\bar{\underline{r}}_n(t-1)-\hat{K}_n(t)\bar{\underline{r}}_n(t-1)+$$

$$\bar{\underline{e}}_{n+1}(t)\underline{r}_n^T(t-1)R_n^{-1}(t-1)\underline{r}_n(t-1)/\beta_n(t)=\bar{\underline{e}}_{n+1}(t)\underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1)$$

Combining terms:

$$\left(\underline{\bar{e}}_n(t)-\hat{K}_n(t)\underline{\bar{r}}_n(t-1)\right)\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)+$$

$$\underline{\bar{e}}_{n+1}(t)\left(\underline{r}_n^T(t-1)R_n^{-1}(t-1)\underline{r}_n(t-1)/\beta_n(t)-\underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1)\right) = 0$$

Substituting from equations A47 and A54:

$$\left(\underline{e}_n(t)-\hat{K}_n(t)\underline{r}_n(t-1)\right)\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)/\left[1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)\right] +$$

$$\underline{\bar{e}}_{n+1}(t)\left(\underline{r}_n^T(t-1)R_n^{-1}(t-1)\underline{r}_n(t-1)/\beta_n(t)-\underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1)\right) = 0$$

Substituting from equation A28 and A47:

$$\left(1-\underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1)\right)\underline{\bar{e}}_{n+1}(t)\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)/\left[1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)\right]$$

$$+\ \underline{\bar{e}}_{n+1}(t)\left(\underline{r}_n^T(t-1)R_n^{-1}(t-1)\underline{r}_n(t-1)/\beta_n(t)-\underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1)\right)=0$$

Multiplying through by $1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)$ and combining terms:

$$\underline{\bar{e}}_{(n+1)}(t)\left(\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)+\right.$$

$$\left.\underline{r}_n^T(t-1)R_n^{-1}(t-1)\underline{r}_n(t-1)\left[1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)\right]/\beta_n(t)-\underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1)\right)$$

$$= 0$$

Since this is true for all $\underline{\bar{e}}_{n+1}(t)$:

$$\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)+\underline{r}_n^T(t-1)R_n^{-1}(t-1)\underline{r}_n(t-1)\left(1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)\right)/\beta_n(t)$$

$$= \underline{k}^{(n+1)^T}(t)\underline{\phi}^{(n+1)}(t-1) \qquad \text{(A56)}$$

Substituting from equation A41 into the right hand side and subtracting $\underline{k}^{(n)^T}\underline{\phi}^{(n)}(t-1)$ from both sides:

$$\underline{r}_n^T(t-1)R_n^{-1}(t-1)\underline{r}_n(t-1)\left(1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)\right)/\beta_n(t)=$$

$$\underline{r}_n^T(t-1)R_{n-1}^{-1}(t-1)\underline{r}_{n-1}(t-1)/V$$

Or: $$\beta_n(t) = V\left[1-\underline{k}^{(n)^T}(t)\underline{\phi}^{(n)}(t-1)\right] \qquad \text{(A57)}$$

For n=0 (since $\underline{\phi}^{(0)} = 0$ from equation 2.4.2):

$$\beta_o(t) = V \qquad \text{*(A58)}$$

This is the same as the equation in Figure 2.4.1 because that equation is for the inputs normalized by the noise so that V=1. For higher orders, substitute A41 into A57:

$$\beta_n(t) = V\left[1-\underline{k}^{(n-1)^T}(t)\underline{\phi}^{(n-1)}(t-1)-\underline{r}_{n-1}^T(t-1)R_{n-1}^{-1}(t-1)\underline{r}_{n-1}(t-1)/V\right]$$

$$= V\left[1-\underline{k}^{(n-1)^T}(t)\underline{\phi}^{(n-1)}(t-1)\right] - \underline{r}_{n-1}^T(t-1)R_{n-1}^{-1}(t-1)\underline{r}_{n-1}(t-1)$$

Substituting for the first term from equation A57 and shifting up one order:

$$\beta_{n+1}(t) = \beta_n(t) - \underline{r}_n^T(t-1)R_n^{-1}(t-1)\underline{r}_n(t-1) \qquad \text{*(A59)}$$

Approximation Used in Equation A53

Equations A37-A39 and A48-A50 are the batch version of the equations for $\underline{k}^{(n)}$ and $\underline{l}^{(n)}$. For the recursive form, the matrices $R^{(n)}$ and $Q^{(n)}$ are given initial values at t=0. In this case, equations A38, A39, A49, and A50 become:

$$R^{(n)}(t-1) = \left[\prod_{j=1}^{t} \lambda(j)\right]R^{(n)}(-1) + \sum_{k=1}^{t}\left[\sum_{j=k+1}^{t} \lambda(j)\right]\underline{\phi}^{(n)}(k-1)\underline{\phi}^{(n)^T}(k-1)/V(k)$$

$$\underline{k}^{(n)}(t) = R^{(n)^{-1}}(t-1)\underline{\phi}^{(n)}(t-1)/V(t)$$

$$Q^{(n)}(t) = \left[\prod_{j=1}^{t} \lambda(j)\right]Q^{(n)}(0) + \sum_{k=1}^{t}\left[\prod_{j=k+1}^{t} \lambda(j)\right]\underline{\phi}^{(n)}(k)\underline{\phi}^{(n)^T}(k)/V(k)$$

$$\underline{l}^{(n)}(t) = Q^{(n)^{-1}}(t)\underline{\phi}^{(n)}(t)/V(t)$$

The equations are initialized at t=0, and data is taken starting at t=1. To make the comparison easier to see, write the expressions for $\underline{l}^{(n)}(2)$ and $\underline{k}^{(n)}(3)$:

$$\underline{k}^{(n)}(3) = R^{(n)^{-1}}(2)\underline{\phi}^{(n)}(2)/V(3)$$

$$= \left[\lambda(1)\lambda(2)\lambda(3)R^{(n)}(-1) + \lambda(2)\lambda(3)\underline{\phi}^{(n)}(0)\underline{\phi}^{(n)^T}(0)/V(1) + \right.$$
$$\left.\lambda(3)\underline{\phi}^{(n)}(1)\underline{\phi}^{(n)^T}(1)/V(2) + \underline{\phi}^{(n)}(2)\underline{\phi}^{(n)^T}(2)/V(3)\right]^{-1}\underline{\phi}^{(n)}(2)/V(3)$$

$$\underline{l}^{(n)}(2) = Q^{(n)^{-1}}(2)\underline{\phi}^{(n)}(2)/V(2)$$

$$= \left[\lambda(1)\lambda(2)Q^{(n)}(0) + \lambda(2)\underline{\phi}^{(n)}(1)\underline{\phi}^{(n)^T}(1)/V(1) + \right.$$
$$\left.\underline{\phi}^{(n)}(2)\underline{\phi}^{(n)^T}(2)/V(2)\right]^{-1}\underline{\phi}^{(n)}(2)/V(2)$$

Comparing these 2 expressions illustrates that $\underline{l}^{(n)}(t)=\underline{k}^{(n)}(t+1)$ if:

(1) $V(t)$ = constant = V

(2) $\underline{\phi}^{(n)}(0) = \underline{0}$

and either:

(3) $\lambda(t) = 1$

(4) $Q^{(n)}(0) = R^{(n)}(-1)$

or:

(3) $\lambda(t)$ = constant = λ

(4) $Q^{(n)}(0) = \lambda R^{(n)}(-1)$

This is the standard set of conditions for the algorithm to be valid. An alternate set of sufficient conditions is:

(1) $V(t)$ = constant = V

(2) $\lambda(t)$ = constant = λ

(3) $Q^{(n)}(0) = \lambda R^{(n)}(-1) + \underline{\phi}^{(n)}(0)\underline{\phi}^{(n)T}(0)/V(1)$

Note that this set allows non-zero initial conditions while preserving the strict equality of equation A53.

A.3 Extension of Derivation

Since $\beta_n(t)$ is the least squares weighting factor (see "Generating K_n and K_n* in section A.2), the equations in figure 2.4.2 are just the Kalman filter version of the equations in figure 2.4.1 (Ref. 20, p. 57), where:

$$P_n(t) \triangleq R_n^{-1}(t) \tag{A60}$$

$$P_n^*(t) \triangleq R_n^{*-1}(t)$$

$$\gamma_n(t) \triangleq \lambda\, \beta_n(t) \tag{A61}$$

Using equations A60 and A61, the equations for $\beta(t)$ (equations A58 and A59) become:

$$\gamma_o(t) = \lambda\, \beta_o(t) = \lambda\, V \tag{A62}$$

$$\gamma_{n+1}(t) = \lambda\, \beta_{n+1}(t) = \lambda\, \beta_n(t) - \underline{r}_n^T(t-1)\, P_n(t-1)\, \underline{r}_n(t-1)$$

$$= \gamma_n(t) - \underline{r}_n^T(t-1)\, P_n(t-1)\, \underline{r}_n(t-1) \tag{A63}$$

Appendix B: Modal Parameterization 3

As described in Ref. 13 and referred to in this thesis as MIMO Parameterization 2, any observable system can be put in the form (assuming the inputs don't affect the outputs directly e.g. no un-integrated accelerometer data):

$$\underline{x}(k+1) = [A_{ij}]\,\underline{x}(k) + B\,\underline{u}(k)$$

where: $$A_{ii} = \begin{bmatrix} 0 & & \\ \vdots & I_{\nu_i - 1} & \\ 0 & & \\ a_{ii1} & \cdots & a_{ii\nu_i} \end{bmatrix}$$

$$A_{ij} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ a_{ij1} & \cdots & a_{ij\nu_{ij}} \end{bmatrix} \quad 0 \cdots 0$$

B = fully populated

$\nu_{ii} = \nu_i$

$$\nu_{ij} = \begin{cases} \min(\nu_j, \nu_i + 1) & j < i \\ \min(\nu_j, \nu_i) & j \geq i \end{cases} \qquad i,j = 1 \text{ to } p$$

Consider a system where the number of outputs equals half the number of states and the outputs are positioned so that $\nu_i = \nu_{ij} = 2$. Then the transformed A_{ii} will be of the form:

$$A_{ii} = \begin{bmatrix} 0 & 1 \\ a_{ii1} & a_{ii2} \end{bmatrix}$$

Suppose further that the sensors were such that $a_{ijk}=0$ for all $i \neq j$. Then the transformed A matrix would be of the form:

$$A = \text{diag} \begin{bmatrix} 0 & 1 \\ a_{ii1} & a_{ii2} \end{bmatrix} \qquad i=1,\ldots,p;\ p=N/2 = n$$

which is precisely the form desired for modal parameterization 3. To get this form, generate the transformation as if the above described situation existed, then apply to the real model. This requires proving that it is possible to choose the sensors as described.

Proof of Existence of Transformation:

Given (see ref. G):

Consider the system $\underline{x}_{k+1} = A\,\underline{x}_k$, $\underline{y}_k = C\,\underline{x}_k$, $C = \begin{bmatrix} \underline{c}_1^T \\ \vdots \\ \underline{c}_2^T \end{bmatrix}$, where A is 2n x 2n. Transform this system using the matrix T:

$$T = \begin{bmatrix} \underline{c}_1^T \\ \underline{c}_1^T A \\ \underline{c}_2^T \\ \vdots \\ \underline{c}_n^T A \end{bmatrix} \qquad T\,A\,T^{-1} = \left[\begin{array}{cc|cc|c} 0 & 1 & 0 & 1 & \cdots \\ a_{111} & a_{112} & a_{121} & a_{122} & \\ 0 & 0 & & & \\ & & \ddots & & \\ a_{211} & a_{212} & & & \\ \vdots & & & & \\ & & & & 0 \quad 1 \\ & & & & a_{nn1} \quad a_{nn2} \end{array}\right]$$

$$\underline{c}_i^T A^2 = \sum_{j=1}^{n} (a_{ij1}\,\underline{c}_j^T + a_{ij2}\,\underline{c}_j^T A)$$

(1) Assume $a_{ijk} = 0$ for $i = j$. Then:

$$\underline{c}_i^T A^2 = a_{ii1}\,\underline{c}_i^T + a_{ii2}\,\underline{c}_i^T A$$

(2) The A matrix satisfies its own characteristic equation so:

$$A^2 - \beta A - \alpha = \text{rank deficient by } \begin{cases} 2 & \text{no repeated roots} \\ >2 & \text{repeated roots} \end{cases}$$

where $\beta = \begin{cases} -\omega_i^2 & \text{flexible mode} \\ \text{- product of 2 exponents} & \text{rigid body mode} \end{cases}$

$\alpha = \begin{cases} -2\,\zeta\,\omega & \text{flexible mode} \\ \text{sum of 2 exponents} & \text{rigid body mode} \end{cases}$

(3) Since $A^2 - a_{ii2}A - a_{ii1}$ is rank deficient by at least 2, two conditions can be imposed on $\underline{c}_i^T$.

(a) Replace one row of $A^2 - a_{ii2}A - a_{ii1}$ with the transpose of the first column of B

(b) Normalize $\underline{c}_i$ so that the largest element of $\underline{c}_i^T A\,B = 1$

If there are repeated roots, or adding the column from B doesn't increase the rank of the matrix, then more conditions could be imposed on $\underline{c}_i$. Any choice which gives T nonsingular is allowed.

The 3(a) and 3(b) conditions give the B matrix in the form required by modal parameterization 3.

References

1. Aoki, M. and R.M. Staley. "On Input Signal Synthesis in Parameter Identification," Automatica, Vol. 6, 1970. p.431.
2. Ayer, F. "Mast Control Experiment," presentation at C.S. Draper Laboratory, Sept. 5, 1984.
3. Bierman, G.J. Factorization Methods for Discrete Sequential Estimation, Academic Press, 1977, Chapters 2 and 3.
4. Bryson and Ho. Applied Optimal Control, 1975. Equations 5.2.7, 5.2.8, 5.2.20, and 5.2.24.
5. Ibid. Equations 6.3.1, 6.3.3, and 6.3.6 (p. 181 and 182).
6. Chen, C.T. Introduction to Linear System Theory, Holt, Rinehart and Winston, 1970. Section 4-3.
7. Chen, Jay-Chung. "Evaluation of Modal Testing Methods," AIAA Dynamics Specialist Conference, Palm Springs, CA, May 17-18, 1984. Paper AIAA-84-1071. p. 561.
8. Citron, S.J. Elements of Optimal Control, Holt, Rinehart, and Winston, Inc., 1969. p.47.
9. Clough, R.W. and J. Penzien. Dynamics of Structures, McGraw-Hill, 1975.
10. Friedlander, B. "Lattice Filters for Adaptive Processing, Proceedings of the IEEE, Vol. 70, No. 8, August 1982, p. 829.
11. Gelb, A. et al. Applied Optimal Estimation, The M.I.T. Press, 1974, p. 1.
12. Ginter, S.D. "Limitations on Active Control of Vibrations in Systems with Clustered Natural Frequencies," M.I.T. PhD thesis, August, 1982.
13. Guidorzi, R. "Canonical Structures in the Identification of Multivariable Systems," Automatica, Vol. 11, 1975, p. 361.
14. Hsia, T.C. System Identification, Lexington Books, 1977. p. 42.
15. Ibrahim, S.R. "A Modal Identification Algorithm for Higher Accuracy Requirements," AIAA Paper No. 84-0928, AIAA Dynamics Specialists Conference, 1984.

16. Juang, J.N. and R.S. Pappa. "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," Journal of Guidance, Control and Dynamics, Vol. 8, No. 5, Sept.-Oct., 1985, p. 620.

17. Kalaba, R. E. and K. Spingarn. "Optimal Inputs and Sensitivities for Parameter Estimation," Journal of Optimization Theory and Applications, Vol. 11, No. 1, 1973, p. 56.

18. Kalaba, R. E. and K. Spingarn. "Optimal Input System Identification for Homogeneous and Nonhomogeneous Boundary Conditions," Journal of Optimization Theory and Applications, Vol. 16, Nos. 5/6, 1975, p. 487.

19. Lee, D.T.L. and M. Morf. "Recursive Square-Root Ladder Estimation Algorithms," IEEE International Conference on Acoustics, Speech, and Signal Processing, 1980. p. 1005.

20. Ljuing, L. and T. Soderstrom. Theory and Practice of Recursive Identification, The MIT Press, 1983. p. 357.

21. Mehra, R. K. "Optimal Inputs for Linear System Identification," IEEE Transactions on Automatic Control, Vol. AC-19, no. 3, June 1974, p. 192.

22. Sinha, N.K. and B. Kuszta. Modeling and Identification of Dynamic Systems, Van Nostrand Reinhold Co., 1983.

23. Sundararijan, N. et.al. "Adaptive Modal Control of Structural Dynamic Systems Using Recursive Lattice Filters," Journal of Guidance, Control, and Dynamics, Vol. 8, No. 2, March-April, 1985, p. 223.

24. Villalba, M.J. "On the Use of Autocorrelation Desampling in High Resolution Spectrum Estimation," M.S. thesis, Aeronautics and Astronautics, M.I.T., June 1984.

25. Wong, E. C. "In-Flight Identification of the Galileo Spacecraft Flexible Mode Characteristics," J. Of Guidance, Control, and Dynamics, Vol. 9, No. 1, Jan.-Feb. 1986, p. 92.

26. "Shuttle Launch Pricing Issues Produces Diverging Opinion," Aviation Week & Space Technology, March 18, 1985. p. 137.