SRC TR 86-69

Design And Real-Time Control Of A Flexible Arm

By

G. H. Frank

SYSTEMS RESEARCH CENTER

UNIVERSITY OF MARYLAND

COLLEGE PARK, MARYLAND 20742

DESIGN AND REAL-TIME CONTROL OF A FLEXIBLE ARM

by

Gilbert Haven Frank

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1986

ABSTRACT

TITLE OF THESIS: Design and Real-Time Control of a Flexible Arm

Gilbert Haven Frank, Master of Science, 1986

Thesis directed by: Prof. P. S. Krishnaprasad
Associate Professor
Electrical Engineering Department

As specialized robotic manipulators have been designed to lift and accurately move small loads, the need for machines with lightweight, low inertia links has been noted. Such designs imply that some structural rigidity must be sacrificed, shifting the mechanical resonance spectrum lower in frequency toward the frequencies of interest to the system controller. These lower resonant frequencies with equal or lower damping cause larger absolute end-effector overshoot, in the absence of compensation schemes.

One method employed to regain system performance, the observer-controller, involves implementation of a control law utilizing feedback from estimates of all the states of a model of the link (plant) including those states not directly measurable. A particular example of the observer-controller based on a reduced order model of a single link flexible arm is developed in this paper.

A plant was designed and built for this experiment consisting of a thin, 36 inch long, aluminum beam connected fixed-free to a motorized

hub. Hub angular position, hub rate, and beam tip acceleration are physically measurable. The plant interfaces with an IBM PC AT through an electronics chassis which also drives the motor.

An observer-controller for the plant resides as a software program in the PC. The model is developed from a modal identification experiment using the beam tip accelerometer as a sensor. The University of Maryland optimization package, DELIGHT.MaryLin, is used to determine the feedback gains required to place the closed loop poles of the plant according to performance constraints specified by the designer.

Test results were recorded using a step input by measuring plant tip motion vs. time and comparing the results with the DELIGHT.MaryLin simulation.

We conclude that the resonant modes of the plant can, to a certain extent, be controlled by this method and that the observer-controller scheme is a viable tool in this context.

ACKNOWLEDGEMENTS

I wish to acknowledge the assistance from all of the people who have supported me during the experiment and during the preparation of this thesis, especially:

Westinghouse Electric Corporation for the generous fellowship award and financial support with the development of the hardware for the experiment.

Professor P.S. Krishnaprasad, University of Maryland, for his guidance in directing the experiment and the preparation of this thesis.

Westinghouse Electric Corporation Executives: John L. Pearson, administrator of the fellowship; Jay G. Fay, George H. McAfee.

Westinghouse Electric Corporation Engineers; Mark J. Ramsey for his extensive mechanical engineering assistance, Clint W. Moulds for his technical advice.

University of Maryland Systems Research Center (SRC) for their support, especially Li-sheng Wang, Michael Fan, and N. Sreenath for their work with DELIGHT.MaryLin and the theoretical design.

The Minta Martin Fund for Aeronautical Research for help with the purchase of the encoder in experimental equipment.

Terri Barnes for her typing.

Table of Contents

I. INTRODUCTION

Recently, as the tasks performed by robotic manipulators have grown beyond the coarsely controlled welder/assembler, specialized equipment has been designed to accurately locate, lift, move and place small loads. Such new designs have no need for the load bearing capability of previous machines and are designed with lighter links to allow faster joint acceleration. Robotic links designed for space use are also lightweight for obvious reasons. These lightweight, low inertia, designs are inherently less rigid, causing a downward shift of the resonant frequency spectrum toward the frequencies of interest to the controller. End-effector time "on the target" is then reduced as the dominant system time constant increases (for the lower resonant frequency) given equal or lower damping ratio. This translates into lost transition time and a higher control cost.

This thesis explores one method, the observer-controller, that may be used to regain lost performance due to reduced link stiffness. A system (plant) was designed and built in order to experiment with a flexible link and then show that a control may be applied which will, to a large extent, negate the flexible nature of the link.

With the observer-controller a control law is utilized which demands that all of the states of the link (plant) are measurable.

Estimates were developed for all of the plant states and these were used in a feedback controller for the hub motor.

The milestones of the experiment are now summarized.

1. Develop the Mathematical Model

The plant must be characterized mathematically. This was done by taking the system transfer function followed by a curve fit operation on the truncated transfer function response. Intuitive knowledge of the plant then allowed rearrangement of the transfer function to specify model states as directly measurable states in the plant. Plant tip acceleration, a measured quantity, was developed mathematically in the model.

2. DELIGHT.MaryLin Optimization

Attention was then turned to the DELIGHT.MaryLin environment and objectives and constraints were specified which the plant must meet for a given step input. One is aware that if a realization [A, b, c] is controllable then one can, with a suitably selected feedback gain, move the eigenvalues of the realization [(A-bk), b, c] to any desired eigenvalue (the pole placement theorem). This fact was used, along with the optimization performed by DELIGHT.MaryLin, to determine the controller gain. The result is a feedback gain matrix which, if all states are available, produces the desired plant performance.

3. Translate to Discrete Time and Develop the Observer-Controller

The open loop plant model and the feedback gain matrix were translated to a discrete time setting utilizing a six term matrix exponential series. The resulting system $[(\Phi - \Gamma K), \Gamma, c]$ was then simulated to assure that it would attain the desired results. At this point the observer gain which will drive the estimated states to the real states was derived. This was done by the use of DELIGHT.MaryLin with modifications by a pole placement algorithm.

4. Implement the Observer-Controller in Software

The Data Acquisition Adapter in the IBM PC allows us to sample the plant analog sensors and the digital sensor (shaft encoder) and to output an error signal to the plant motor with simple subroutine calls. All of the measured plant data is then available to a FORTRAN routine which recursively calculates the "future" (next) states of plant model in real time. The "present" states then drive the feedback gain matrix to develop the plant error signal.

5. Testing, Discussion of Results, Conclusions

These follow in sufficient detail.

II. MECHANICAL HARDWARE

The plant described in this chapter was specifically designed to support experimental investigation of the performance of an observer-controller used with a flexible link.

The flexible link, which is the subject of the investigation, was specified first. Here, the intent was to select a link which produced at least two resonant modes well below the Nyquist frequency of the controller. Beam tip lag (w.r.t. a rigid beam) of at least one inch under full motor acceleration was also deemed appropriate. These are not the specifications one would want in a link (in fact higher frequency modes and less tip motion are desired), but they describe a system which can be difficult to control and interesting for this experiment.

To determine the plant's resonant modes, routine calculations using mechanical engineering beam tables and textbook resonance formulas led to a choice of a rectangular cross-section aluminum beam with a fundamental fixed-free resonance of approximately 2.7hz. The lower frequency fixed-free resonant modes (i.e. those modes where significant mass decouples from the rigid system) are modified when the fixed end of the beam is secured by a less than infinite mass. The plant is a system where the beam is clamped onto the joint shaft and the shaft is free to rotate. The joint inertia is somewhat higher than ideal, because a 10 inch diameter table is built into the joint shaft to be used as a platform for future experiments. The inertia of the plant shaft and table (hub) is 10% of the inertia of the rigid system, with the net result that the first plant resonant frequency is about 9hz. The lumped model calculation of the first resonance is 10.3hz. The results follow, and the development is found in appendix A.

Given: Appendix A and

w_1 = first fixed-free bending mode frequency of the beam

w_2 = first bending mode of the experimental system

α, β = numerator terms

<u>for the fixed-free beam</u>

$$\theta_2/\theta_1 = \alpha/((s^2/w_1^2) + 1) = \alpha/((s^2/(\sqrt{k/.81})^2) +1)$$

<u>for the experimental system</u>

$$\theta_2/T_1 = \beta/((s^2/w_2^2) +1) = \beta/((s^2/(\sqrt{k/.212})^2) +1)$$

Beam measurement shows that $w_1 = 2.7(2\pi)$ rad/sec. The experimental system resonance is approximated by 2.7 (.81/.212) = 10.3hz.

Once the beam and hub were chosen, it was important to consider the torsional stiffness requirement for the joint shaft, so that additional resonances were not introduced. A reasonable specification is ten times the frequency of the highest controllable resonance. There are two stiffnesses to consider, that of the "lower" shaft (k_1) connection to the encoder and the "upper" shaft (k_2) to the table and beam.

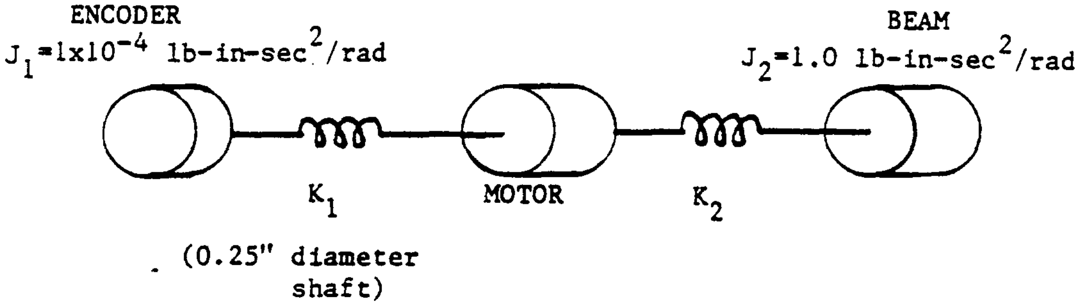

ENCODER
J_1=1x10^{-4} lb-in-sec^2/rad
BEAM
J_2=1.0 lb-in-sec^2/rad
K_1
MOTOR
K_2
(0.25" diameter shaft)

Figure 1 - Joint Shaft Resonance Model

To attain a resonant frequency of 200 hz, with the inertial load of 1.0 lb-in-sec^2/rad, the upper aluminum shaft (with spring constant K_2) must have a stiffness of 1.5 x 10^6 pound-inch/radian. This implies a diameter greater than 0.75 inch. The lower shaft must be reduced to 0.25 inch diameter to mate with the encoder. This diameter is adequate to keep the resonant frequency above 200 hz with the encoder rotor load.

The motor selected for this experiment is a brush-type dc torque motor using samarium-cobalt permanent magnets. The peak output torque is 40 pound-inches and maximum no-load speed is 14 radians/second. Calculations were performed to insure that full torque applied in step fashion to the beam would cause an initial beam tip lag (assuming first mode bending only) of 2 inches w.r.t. the rigid beam. Both the motor and the tachometer are manufactured by Inland Motors. Specific data is shown on the manufacturer's data sheets in the appendix.

Colocated on the joint shaft with the motor is a brush type dc tachometer and an absolute reading optical position encoder. The tachometer develops 0.12 volts dc/radian/second of joint shaft rotation. Because the tachometer is sensitive to output signal corruption from the magnetic field produced by the motor, it must be located at a safe distance. Designing the shaft to be too long will accentuate its radial runout and, so, compromise must be made. The generated magnetic field is especially severe in motors utilizing the high energy rare-earth magnets which narrowly focus the magnetic tooth (compared to an Alnico magnet design).

The digital encoder rotates with the joint shaft via a bellows coupling connection. The resolution (and accuracy) of the encoder is 12 bits, allowing one to discern as little as 0.088 degree of shaft angular motion. The encoder is a fully housed unit with self-contained

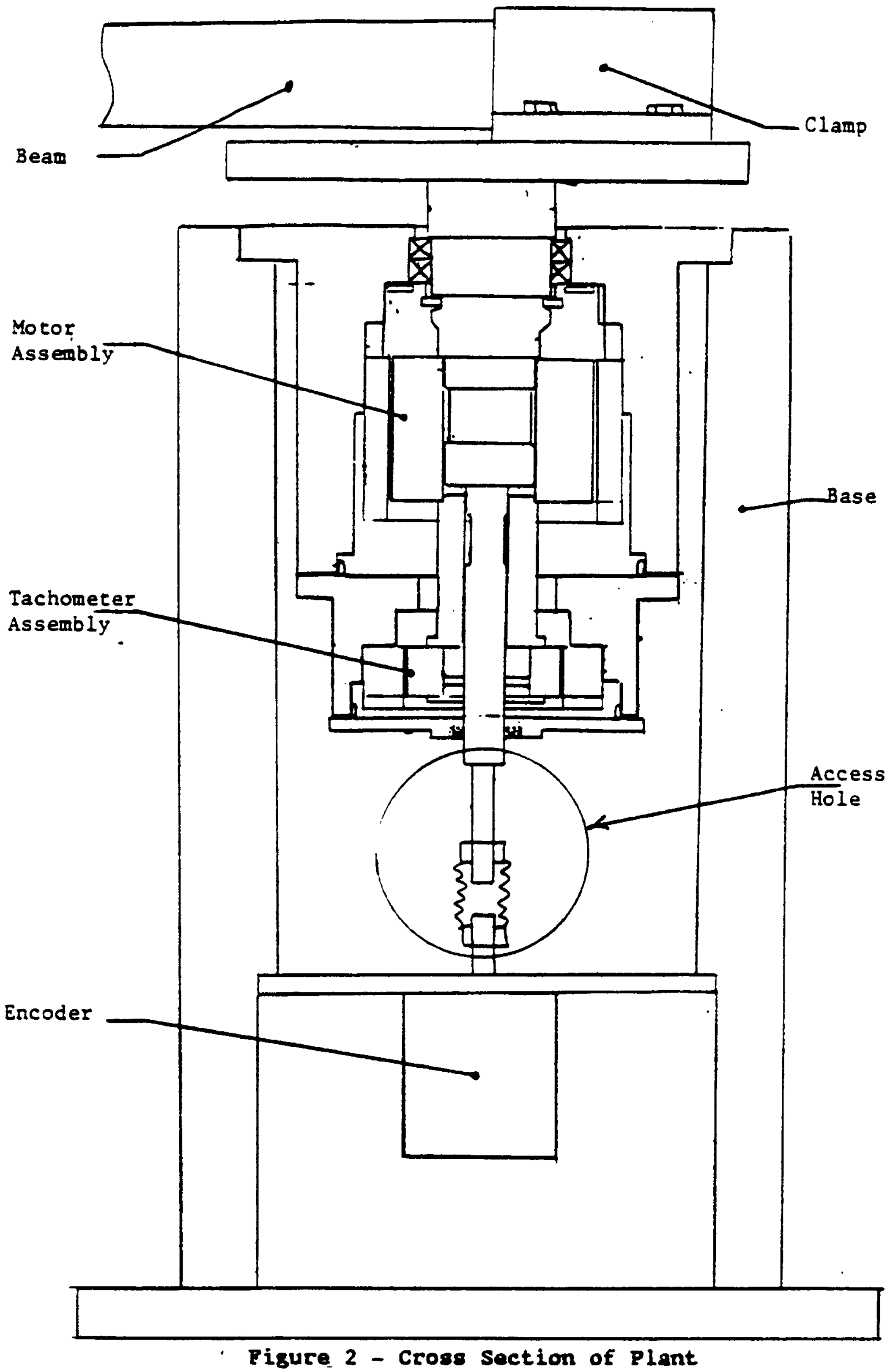


Figure 2 - Cross Section of Plant

electronics generating a serial digital natural binary output from the gray code developed by the glass encoder wheel. The encoder shaft is coupled to the joint shaft with the bellows coupling to forgive small radial and axial misalignments. The encoder is manufactured by BEI Motion Systems Company.

The third plant sensor is a Sunstrand Mini-Pal 2180 pendulous accelerometer which is mounted in a cable clamp at the tip of the beam. The frequency response of the accelerometer is essentially flat from dc to over 1000hz. In order to control tip motion of the beam, tip motion must be sensed to drive the estimated states of the beam to the plant states. The accelerometer is the only sensor in the rotating frame, and its output is responsible for "correcting" the four beam states used in our model of the plant. The sensitive axis of the accelerometer is adjusted to read only the acceleration component normal to the tip of the beam in the plane of tip motion as the hub is rotated. Differential tip acceleration (relative to the rotating frame) is generated in the analog electronics assembly.

An aspect of considerable concern in the design of the joint is the bearing configuration. An adequate design must virtually eliminate shaft radial and axial movement while retaining minimum friction and must allow for shaft expansion from motor heating. This balance was achieved with a duplex preloaded pair of bearings between the motor and beam and a single bearing between the tachometer and encoder providing only radial support. The preload was adjusted during assembly by changing the length of a spacer (sanding the surface until proper fit was achieved). The relatively large spacing between the bearings allows for the highly unbalanced load inertia and resists the overturning moment.

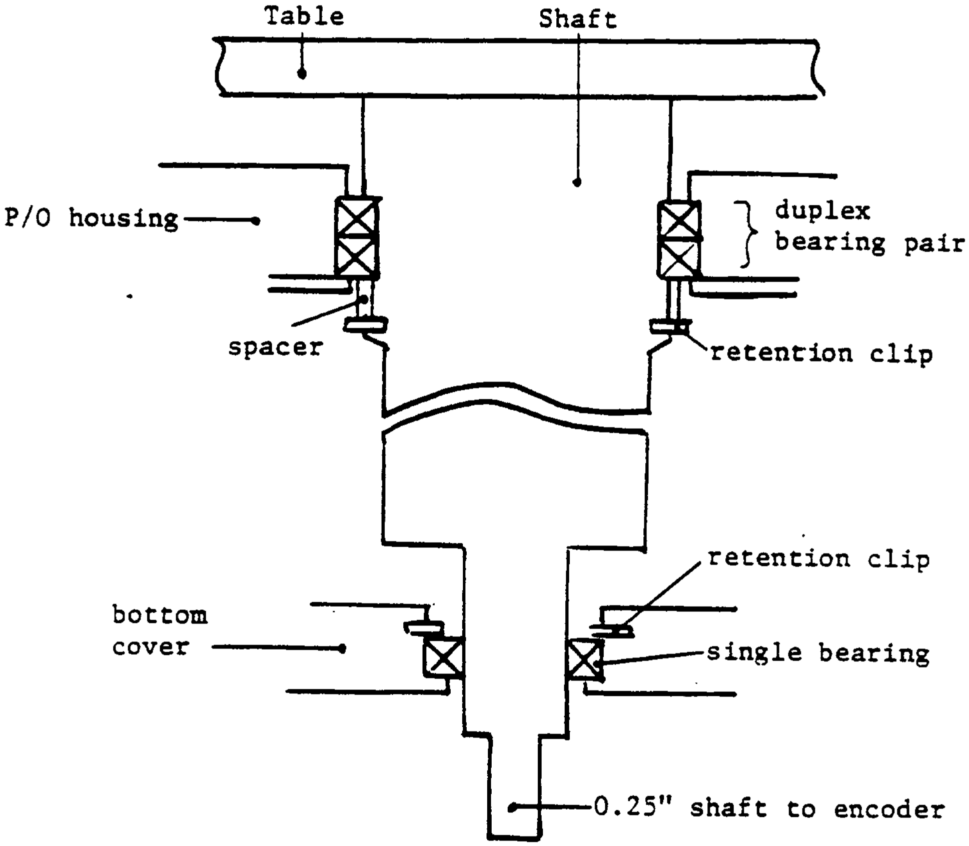


Figure 3 - Functional View of Shaft Bearing Design

Clearances are approximately $\leq$ 0.0005 inches bearing outer diameter (o.d.) to housing and $\leq$ 0.001 inch bearing inner diameter (i.d.) to shaft.

Joint shaft to housing concentricity must be maintained in order to hold a nominal 0.005 inch motor and tachometer air gap clearance. Eccentricity greater than several thousandths of an inch cause the motor torque gain to be position sensitive and can cause rotor to stator interference as the rotor heats during operation.

Access to the internal components for assembly and repair must be a factor in the design. A permanent magnet torque motor of this type must be installed with the rotor and stator nested to complete the magnetic circuit and retain the designed magnetization. The rotor fits over a tightly controlled shaft diameter and the stator fits within a tight bore. As the motor stack length is two inches this is a difficult task requiring open access for assembly. For this reason, the housing is designed in three sections (base, motor with duplex bearing, tachometer with single bearing). The bolt pattern between the motor and tachometer sections is symmetrical so that relative rotation of the sections during assembly can balance any small eccentricity of the shaft and housing. Refer to figure 2 for details.

III. ELECTRICAL INTERFACE

The electrical interface chassis is a custom designed assembly containing the necessary circuits to allow communication between the IBM PC and the plant. The chassis contains power supplies and has a fan for cooling. Functional connections to the interface chassis are shown in Figure 1.

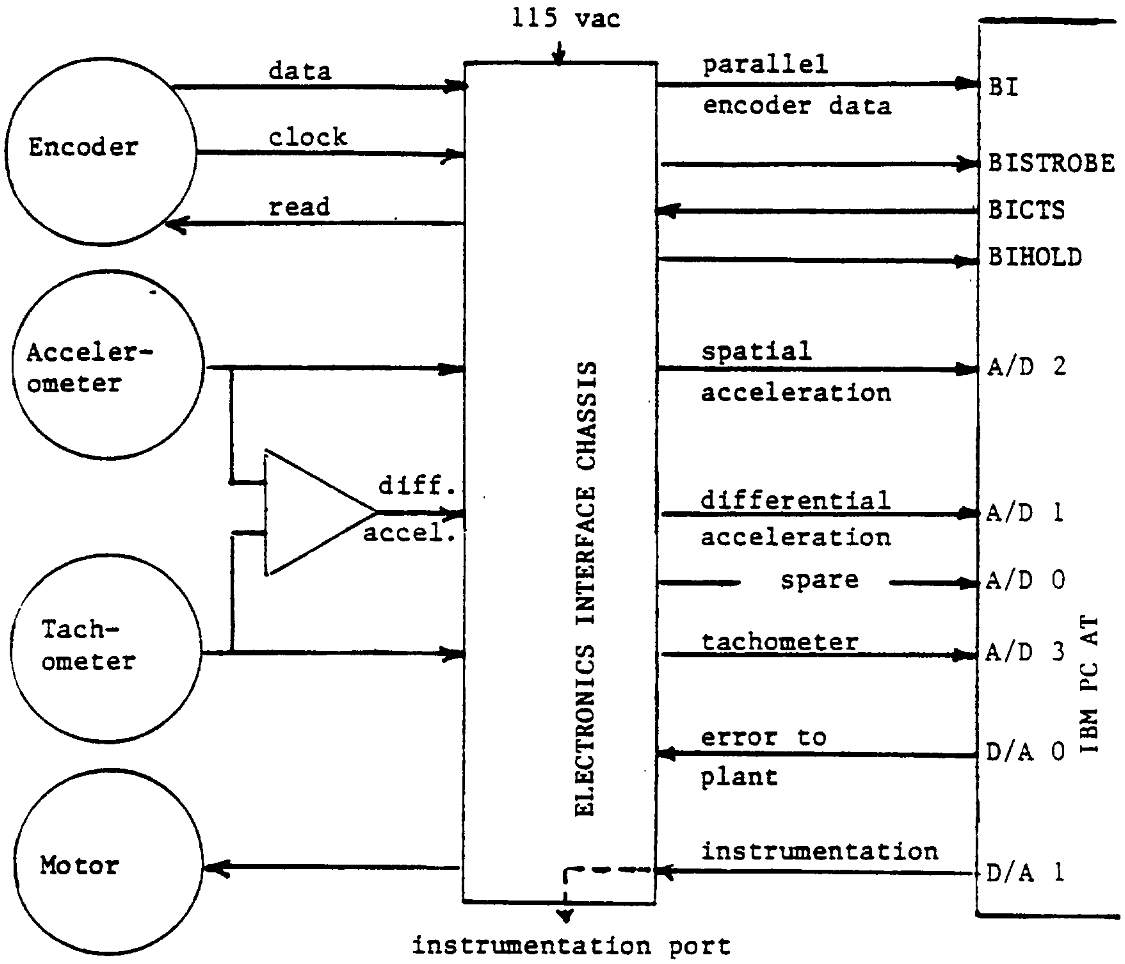


Figure 1 - Electronics Interface Chassis Block Diagram

Three power supplies are included in the chassis. The +5 volt dc supply for the digital logic is a switching regulator with foldback current limiting capable of 5.0 amperes output and is manufactured by Lambda. The modular linear +/-15 volt supply delivers up to 150MA to the load and is manufactured by Technetics. The unregulated supply used to power the motor amplifier is a discrete design utilizing a Stancor power transformer, a full-wave diode bridge block and a single 2100 uf. electrolytic filter capacitor. The primaries of all of the supplies as well as the secondary of the power amplifier are fuse protected.

The power amplifier for the plant motor is of the "H" bridge configuration. It is a four-quadrant design allowing acceleration and braking for either clockwise or counter-clockwise rotation. The output stage is electronically current limited to avoid motor demagnetization during high speed rotation reversal (plugging), where the motor's back emf may add to the amplifier's output voltage and exceed the rated motor current.

The power amplifier is designed with voltage feedback from the bridge output to the input summing amplifier to control the forward voltage gain of the amplifier. The voltage loop bandwidth is approximately 3 Khz, and the current limit is set for 1.5 amperes. Current limit is implemented by removing power from the first stage of the Darlington output driver. A simple lag corner at the data rate of the PC is included ahead of the power amplifier to smooth the transitions produced by the D/A converter.

The plant analog sensors, the shaft tachometer and tip accelerometer employ analog prefilters to adjust their gain and frequency characteristics. The accelerometer is scaled to produce 0.38 volts/g with a simple lag at 710hz. The tachometer is scaled to develop 0.408 volts/radian/second with a simple lag at 2.3Khz. Differential acceleration is also scaled to produce 0.38 volts/g but with a critically damped quadratic roll off at 110hz.

Since the analog signals are available continuously at the interface to the IBM PC Data Acquisition Adapter, no special handshaking is needed. When the program resident in the PC requires an analog input, the program calls the specific input in the data acquisition adapter. The analog value at that time is A/D converted and returned to the calling program as an integer value between 0 and 4095. Negative 10.0 volts at the input is read as integer 0, zero volts as integer 2048, and positive 9.995 volts as integer 4095.

The outputs, either error to the plant or a value to the instrumentation port, are scaled similarly. Integer 2048 in the program produces 0.0 volts at the output at the D/A converter (zero order hold).

The binary interface is complicated by the fact that it must connect the encoder (synchronous due to its self-contained clock) to the PC which is asynchronous because the program execution time line varies slightly due to nonconstant execution time and I/O calls, i.e., breathes. A handshaking scheme is employed requiring the following signals:

1. BISTROBE (binary input strobe) from the electronics chassis to the PC to signal that valid data is on the bus.

2. BICTS (binary input clear to send) from the PC to the electronics chassis to signal that the PC will read the bus unless BIHOLD is set.

3. BIHOLD (binary input hold) from the electronics to the PC to signal that the bus data is or may be invalid.

4. CONDREAD (conditional read) from the electronics to the encoder to signal the encoder to send a serial word to the electronics.

5. LATCH (inside electronics) 16 μs earlier and overlaps CONDREAD.

The BIHOLD is used to stop data transfer in the event that there is a race condition between BISTROBE and BICTS. If BICTS arrives before LATCH goes down, the PC has ≥ 16 μs to read the data word. If

BICTS arrives during the time LATCH is down but before READ goes down, then serial data transfer from the encoder is inhibited and the PC reads the last parallel data word. If BICTS somehow arrives while READ is low, the PC is put into a WAIT STATE with BIHOLD until new valid parallel data is available (16 μs maximum wait time).

The output of the serial to parallel converter (encoder position word) can be offset bit by bit by the binary addition of "1" from an open DIP switch or a "0" from a closed switch. This has the effect of adjusting the zero of the encoder by varying the DIP switches. The encoder position may be monitored by an octal LED display.

The binary encoder output is then scaled in the PC by 1/16 to balance the binary connection to the PC. The binary input is connected between bit 15 and bit 4 with bits 0-3 grounded. Within the PC, the output is integer -2048 to +2047 corresponding to -180 degrees to +179.91 degrees.

IV MODAL IDENTIFICATION AND MODEL DEVELOPMENT

In this step of the experiment, the plant is mathematically characterized empirically by measurement of the physical plant. The results of the mathematical model will be translated to discrete time via a matrix exponential expansion to design the controller.

Two plant experiments must be performed to generate transfer functions from D/A (IBM PC) output to shaft position and to beam tip acceleration. Both experiments were performed using the random noise output of a Hewlett-Packard 3582A Spectrum Analyzer (which was used to generate the transfer function) then, using the continuously swept (stepless) sinusoidal output of an H-P 3314 and the display of the 3582A. The swept-sine source produced a transfer function with better low frequency coherence and, so, this data was used. The displayed transfer functions were then dumped onto an X-Y Plotter to save the results for the curve fitting process.

To keep the model order as low as possible, the transfer functions were not curve-fitted blindly by a computer utility. While this procedure could produce a better average fit, it was felt that intuitive knowledge of the plant could be used during curve-fitting to obtain the most efficient, i.e., minimum order representation of the plant.

Figure 1 is the D/A output to plant shaft position frequency response. From this figure, the transfer function (truncated past 10hz) is developed in Equation 1.

$$\frac{\theta_{shaft}}{V} = \frac{0.86\left[\frac{s^2}{(2\pi x2.7)^2} + \frac{2(.025)s}{2\pi x2.7} + 1\right]}{s\left[\frac{s}{2\pi x1.16} + 1\right]\left[\frac{s^2}{(2\pi x9)^2} + \frac{2(.13)s}{2\pi x9} + 1\right]}$$

The adjusted load corner was found from this transfer function to be 7.3 rad/sec (using only shaft and table inertia produces 21 rad/sec). This can be explained by the increase in load inertia due to the portion of the beam which is moving rigidly with the motor shaft during bending modes. This development may be found in the appendix A.

Figure 2 gives the D/A output to beam tip acceleration frequency response. From this figure, the transfer function (truncated past 20hz) is developed in Equation 2.

$$\frac{\ddot{y}}{V} = \frac{0.125\ s\left[\frac{s^2}{(2\pi x9.5)^2} + \frac{2(1)s}{2\pi x9.5} + 1\right]}{\left[\frac{s}{2\pi x1.16} + 1\right]\left[\frac{s^2}{(2\pi x9)^2} + \frac{2(.11)s}{2\pi x9} + 1\right]\left[\frac{s^2}{(2\pi x20.6)^2} + \frac{2(.02)s}{2\pi x20.6} + 1\right]}$$

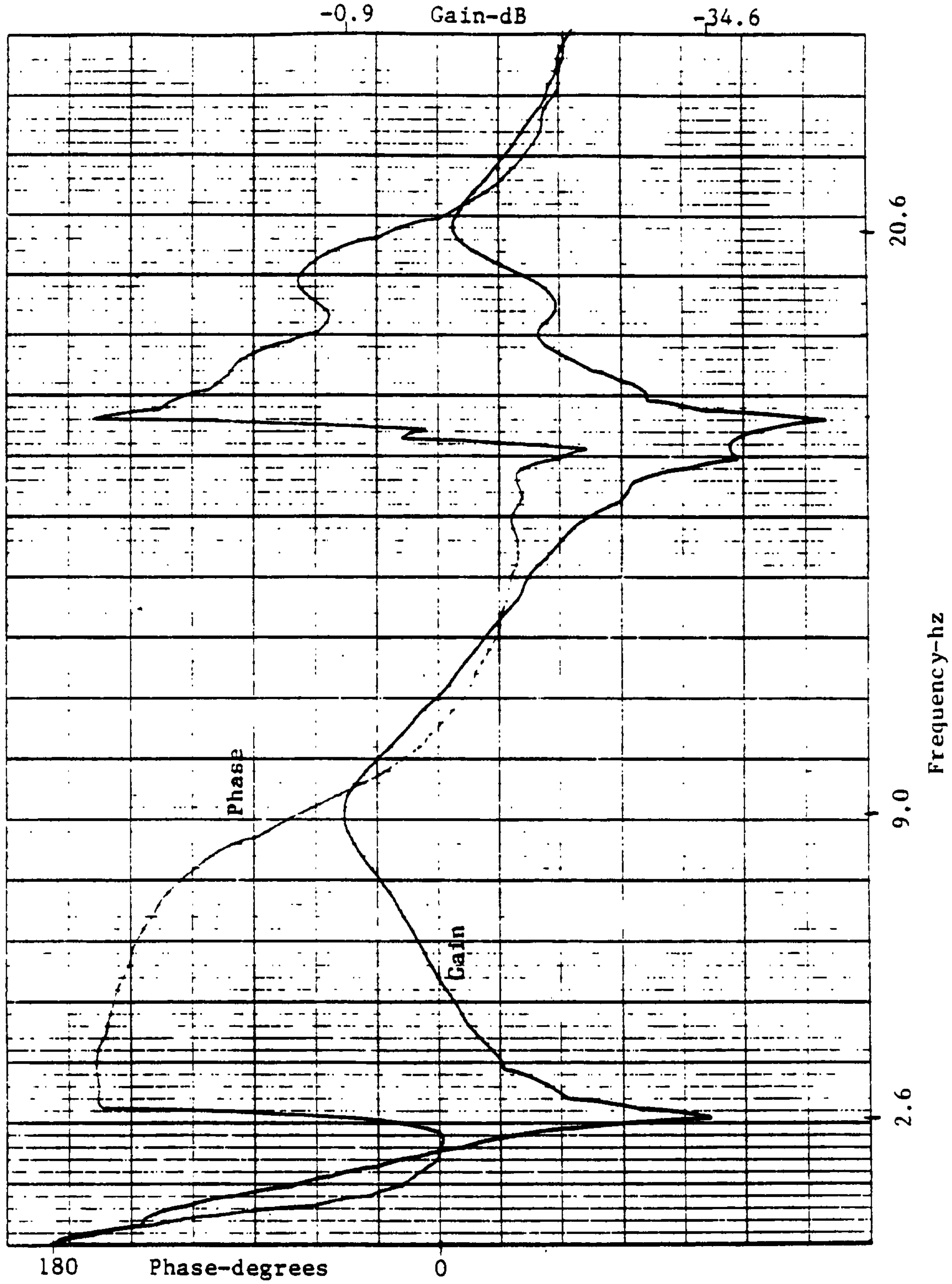


Figure 1. D/A Output to Plant Shaft Position Frequency Response

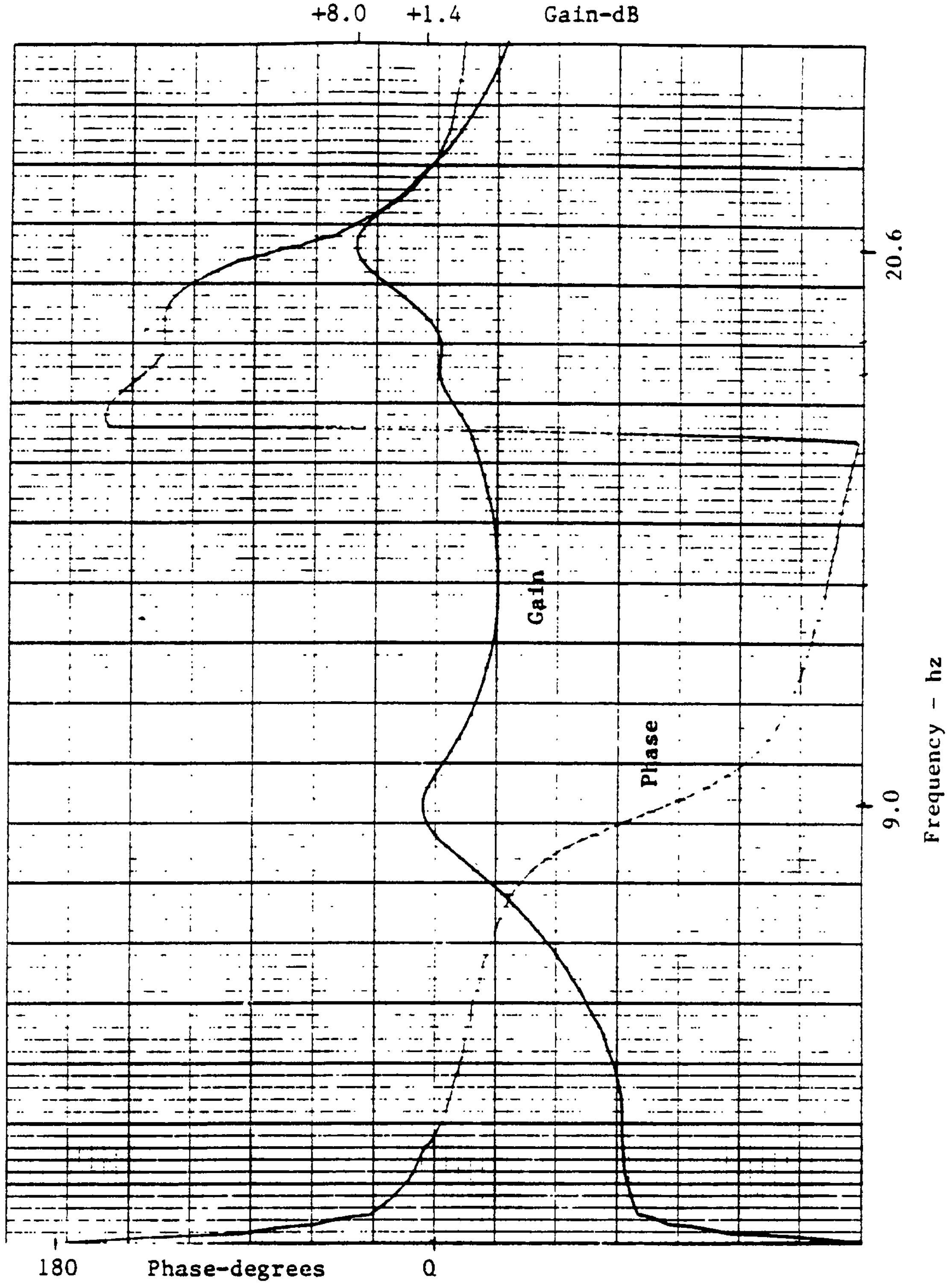


Figure 2. D/A Output to Beam Tip Acceleration Frequency Response

(0 - 25 hz)

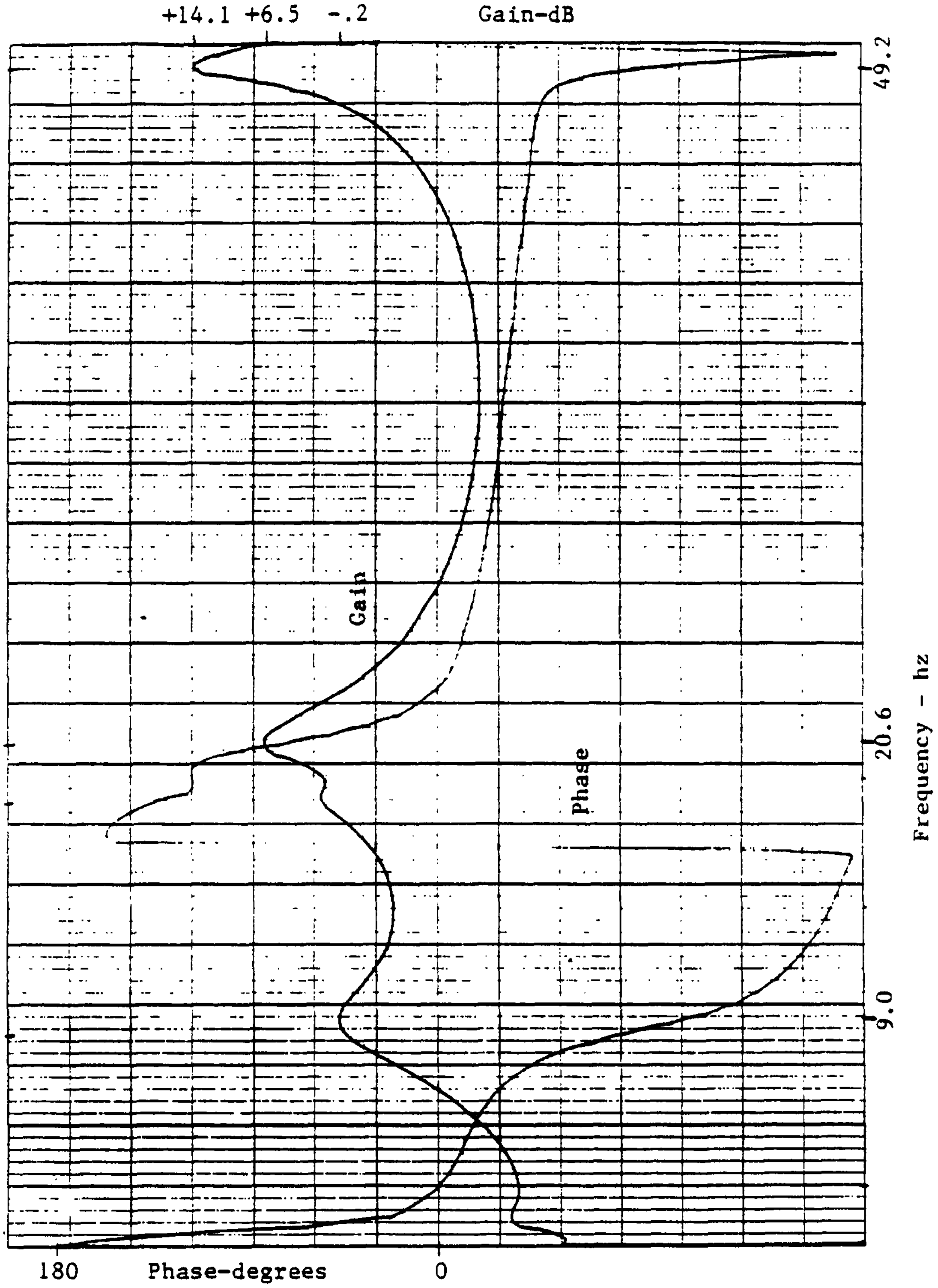


Figure 3. D/A Output to Beam Tip Acceleration Frequency Response (0 - 50 hz)

Equation 2 is the transfer function that is used to model the plant. Figure 3 is an extended frequency response of D/A output to tip acceleration. There are no resonant modes between 20.6 and 49.2hz and the model was, thus, truncated after the 20.6 hz mode.

An estimate of the PC speed was made using a sample program to determine the frequency at which the model must be truncated. The result of this test was a goal of 100hz iteration rate which will alias any frequency greater than 50hz. The ratio of 100/20.6 ≈ 5 samples per cycle (phase shift of (20.6/50) 180° =72 degrees) does not adequately describe the 20.6hz mode but efforts are being made to increase the speed of the data acquisition adaptor. Because the 20.6 hz mode was very strong it was retained in the mathematical model.

The actual curve fitting was accomplished knowing the asymptotic nature of the voltage to tip acceleration must be of the form:

$$\frac{\ddot{y}}{V} = \frac{K_1}{s\left[\frac{s}{w_1}+1\right]} \; \frac{K_2 \prod_{i=1}^{m} \frac{s^2}{a_i^2} + 1}{\prod_{j=1}^{n} \frac{s^2}{b_j^2} + 1} \; \frac{K_3 \, s^2}{1}$$

Referring to the exact y/v transfer function in Equation 2, one may put the transfer function in the above form. The intent is to preserve the plant states (sensors) as model states. Observing Equation 4.

$$\frac{y}{V} = \underbrace{\frac{2.5}{s\left[\frac{s}{7.3} + 1\right]}}_{\text{MOTOR}} \; \underbrace{\frac{0.15664\left[\frac{s^2}{(59.69)^2} + \frac{2s}{59.69} + 1\right]}{\left[\frac{s^2}{(56.55)^2} + \frac{.22s}{56.55} + 1\right]\left[\frac{s^2}{(129.43)^2} + \frac{.04s}{129.43} + 1\right]}}_{\text{BEAM}} \; \underbrace{\frac{.38s^2}{1}}_{\text{ACCELEROMETER}}$$

$$= \frac{b_1 s^5 + b_2 s^4 + b_3 s^3 + b_4 s^2 + b_5 s + b_6}{s^6 + a_1 s^5 + a_2 s^4 + a_3 s^3 + a_4 s^2 + a_5 s + a_6}$$

A cascade observer-canonical representation is developed in figure 4.

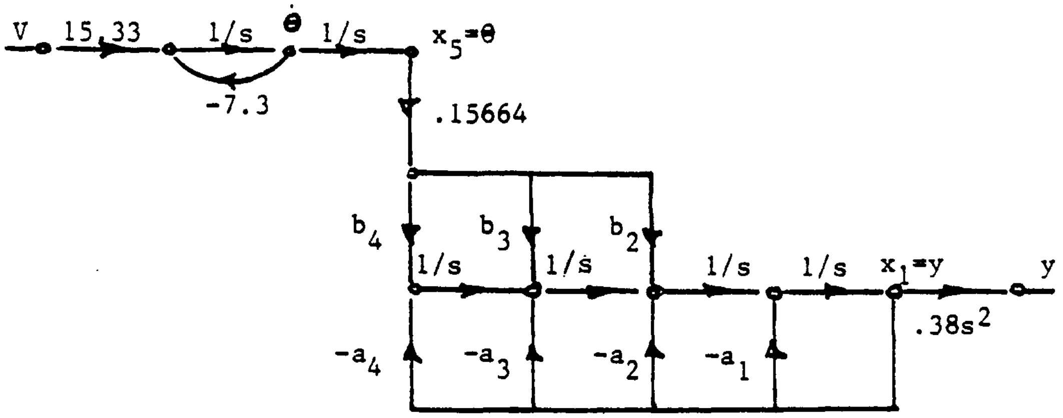


Figure 4 - Plant in Canonical Form

The double differentiation may be replaced by the following:

$$\ddot{y} = [(a_1^2-a_2) \ -a_1 \ 1 \ 0 \ b_2 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} (.38)(.15664)$$

The derivation may be found in appendix B.

The Fortran program to close a low bandwidth position loop around the plant to hold the beam on position so that the transfer functions may be taken is found in appendix C.

V. DELIGHT.MaryLin DESIGN

DELIGHT.MaryLin is an interactive, optimization-based, computer-aided design system which can be employed to assist in the development of feedback control systems. Knowledge of the mathematical representation of the system and that the system is controllable is needed in order to use DELIGHT.MaryLin to optimize a set of specified goals. For this experiment DELIGHT.MaryLin is allowed to adjust a state feedback gain matrix to optimize the system's closed loop poles, achieving the specified performance.

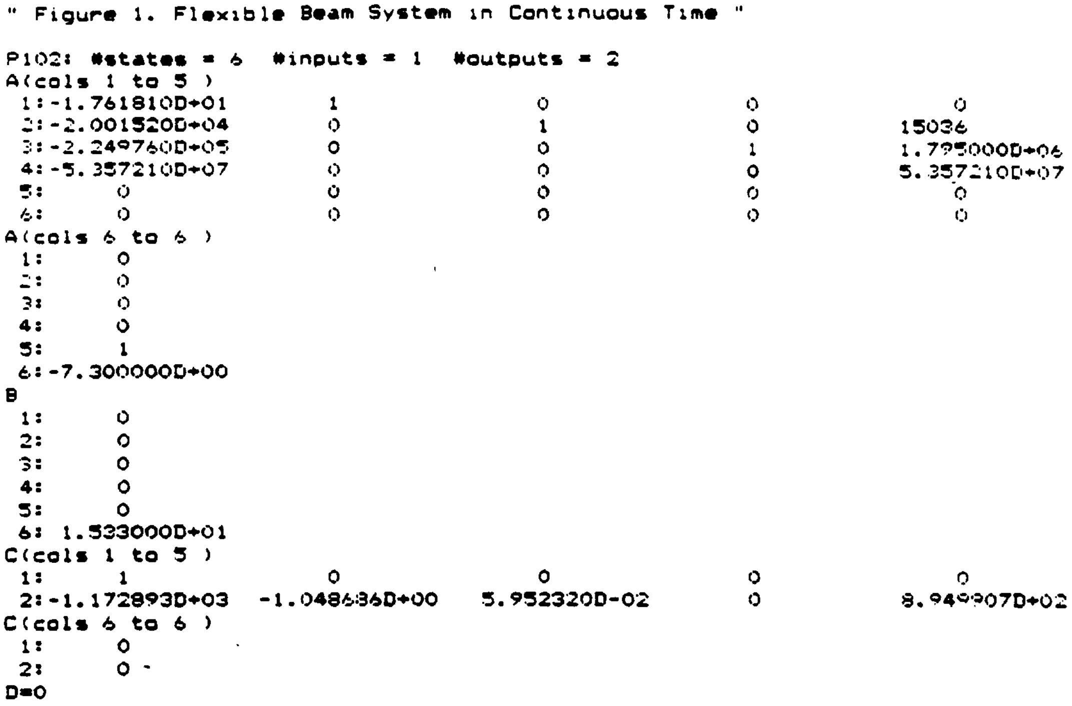
A significant advantage of DELIGHT.MaryLin is the level of parameter control which a designer may exert by communicating with the process to fine tune the resulting system. Development of hard constraints (such as, "the system must be stable"), soft constraints allowing some flexibility (such as, "overshoot should be ≤ 10%") and objectives (such as, "minimize tip acceleration") are allowed. The advantage is that numerical design requirements along with the designer's intuition and experience combine to produce the resultant system.

As the process runs, progress may be observed by defining any plant variable to DELIGHT.MaryLin along with the performance goals for that parameter and then one may view the performance comb (Pcomb) for that variable. The Pcomb shows absolute and relative performance w.r.t. good and bad limits set with the performance goals. We may specify a single point or continuous good/bad loci. Specifications may compete in the optimization process but hard constraints are satisfied first, soft constraints next, and objectives last. If there is more than one constraint or objective within a class, they are weighted equally until one is satisfied. Simulation design parameters (in our case, the feedback matrix elements) must be weighted by use of variation statements to insure equal attention from the algorithm.

Information is passed to DELIGHT.MaryLin by files. The only required file, the Setup File (S) must contain the system matrix sizes and elements, variable assignments, design parameter declarations, and simulation outputs. Other files are optional. A description of the files used in this experiment follows. The Design Parameter File (P) lists starting values for the declared design parameters (and variation statements). The functional Multicost File (FM) contains performance objectives. The Functional Inequalities File (FI) contains the performance constraints (hard and soft).

To use DELIGHT.MaryLin , the system transfer function y/v developed in chapter IV is expressed in state space form.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + bu(t), \quad x(0-) = 0 \\ y(t) &= cx(t) + du(t), \quad d = 0 \end{aligned} \qquad (1)$$



```
" Figure 1. Flexible Beam System in Continuous Time "

P102: #states = 6  #inputs = 1  #outputs = 2
A(cols 1 to 5 )
 1:-1.761810D+01       1            0            0            0
 2:-2.001520D+04       0            1            0         15036
 3:-2.249760D+05       0            0            1         1.795000D+06
 4:-5.357210D+07       0            0            0         5.357210D+07
 5:      0             0            0            0            0
 6:      0             0            0            0            0
A(cols 6 to 6 )
 1:      0
 2:      0
 3:      0
 4:      0
 5:      1
 6:-7.300000D+00
B
 1:      0
 2:      0
 3:      0
 4:      0
 5:      0
 6: 1.533000D+01
C(cols 1 to 5 )
 1:      1             0            0            0            0
 2:-1.172893D+03  -1.048636D+00  5.952320D-02    0         8.949907D+02
C(cols 6 to 6 )
 1:      0
 2:      0 -
D=0
```

Figure 1 - Exact Representation of Equation (1)

In our A,b,c model above, care has been taken to make the model's states and plant's states, where possible, the same to provide measurable feedback for the experiment.

The A,b,c coefficients are developed in the S file and include the input (a step function with value 0.3 at time 0^+), the design parameters (feedback gain constants for each of the six model states), the desired outputs (tip position, tip acceleration and shaft position) and the step bounds (constraint on tip position).

Engineering specifications are now developed to define how the system is to respond to a step input. The object of this experiment is to take a system with a flexible link (beam) and control the bending modes of the link so as to control the tip of the beam as the tip moves through a step 0.3 radian angular change. To control the tip in the sense of minimizing overshoot and minimizing slew time, competing goals must be specified. Control of higher frequency resonant modes (tip vibration) during beam slewing and settling phases is also desired. This relates to minimizing tip acceleration. To apply these goals to the DELIGHT.MaryLin environment, the beam tip position in terms of the maximum and minimum position envelope allowable over a given time is considered. Thus the good loci (upper and lower limits) is defined by using the step bounds in the S file.

"Step_bounds" defines the limits shown at the end of this chapter in the Ytr(1) plots. Slew Time is defined as the time ($\succ$ 1.8 second) when the beam moves from 0 to 0.3^{+} radian, overshooting as allowed by the constraint file (FI); Settle Time is defined as the time ($\prec$ 1.8 second) when the beam reverses direction and settles into the commanded angle, undershooting this position as allowed in the FI file. Slew limits are +/-5% while settle limits are +/-2% of the step amplitude.

For the P file, an initial estimate for the design parameters is needed. A close estimate is required to enable the DELIGHT.MaryLin gain search algorithm to converge, and this was obtained via a solution of the Algebraic Riccati Equation (ARE) using the program Classical Control (CC). The ARE was solved with different estimates for the state weighting matrix, Q, checking the eigenvalues of the resulting system (A-bk) in each case, until a sufficiently stable solution was reached. The CC printed result of the system A matrix, Q matrix, and initial gain matrix, K is shown in Figure 2.

Through the FI file performance is specified w.r.t. the step bounds in the S file. A soft constraint is defined to keep the locus within the bounds, i.e., this is "good". Allowing for some inexactness of solution "bad" is specified as 10% above and below the bounds.

To minimize beam tip acceleration, the FM file is used, allowing more acceleration during the slew time than during the settle time. The actual objective limits are somewhat of an educated guess in that to achieve rapid tip response, more acceleration must be allowed during slew, but motion must be rapidly damped when slew is completed. Figure 3 depicts the desired tip acceleration profile.

```
" Figure 2. Beam System A matrix "

P1: #states = 0  #inputs = 6  #outputs = 6
(cols 1 to 5 )
 1:-1.761810D+01        1              0              0              0
 2:-2.001520D+04        0              1              0          15036
 3:-2.249760D+05        0              0              1          1.795000D+06
 4:-5.357210D+07        0              0              0          5.357210D+07
 5:      0              0              0              0              0
 6:      0              0              0              0              0
(cols 6 to 6 )
 1:      0
 2:      0
 3:      0
 4:      0
 5:      1
 6:-7.300000D+00

" Figure 2. State Weighting Matrix, Q "

P103: #states = 0  #inputs = 6  #outputs = 6
(cols 1 to 5 )
 1:  1000           0              0              0              0
 2:      0          0              0              0              0
 3:      0          0              0              0              0
 4:      0          0              0              0              0
 5:      0          0              0              0              0
 6:      0          0              0              0              0
(cols 6 to 6 )
 1:      0
 2:      0
 3:      0
 4:      0
 5:      0
 6:      0

" Figure 2. Riccati Solution Matrix, P "

P106: #states = 0  #inputs = 6  #outputs = 6
(cols 1 to 5 )
 1: 1.211522D+02   1.151681D-02  -7.462151D-03  -4.410780D-06  -9.592389D+01
 2: 1.151681D-02   7.667268D-03   9.132982D-07  -5.823677D-07  -1.108217D+00
 3:-7.462151D-03   9.132982D-07   5.512222D-07   2.655332D-10   5.847051D-03
 4:-4.410780D-06  -5.823677D-07   2.655332D-10   7.431679D-11   1.572703D-04
 5:-9.592389D+01  -1.108217D+00   5.847051D-03   1.572703D-04   4.280860D+02
 6: 6.530821D-01  -9.900086D-03  -8.816153D-05   1.503254D-06   4.383260D+00
(cols 6 to 6 )
 1: 6.530821D-01
 2:-9.900086D-03
 3:-8.816153D-05
 4: 1.503254D-06
 5: 4.383260D+00
 6: 1.645590D-01

" Figure 2. Starting Gain Matrix, K "

P107: #states = 0  #inputs = 6  #outputs = 1
(cols 1 to 5 )
 1: 1.001176D+01  -1.517682D-01  -1.351515D-03   2.304485D-05   6.719550D+01
(cols 6 to 6 )
 1: 2.522692D+00
```

Figure 2 - Derivation of DELIGHT.MaryLin Starting Point

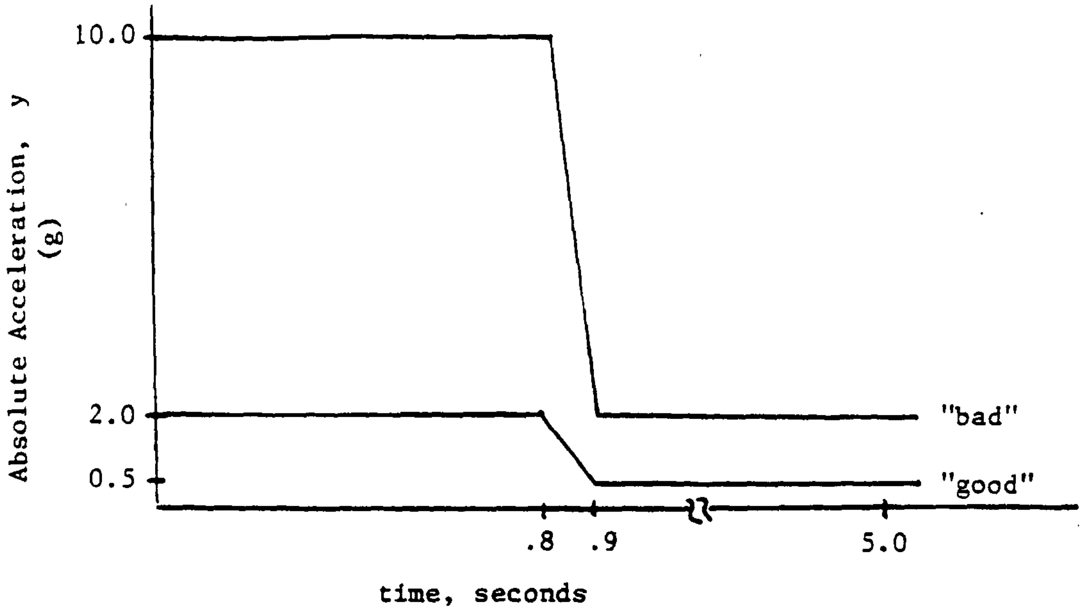


Figure 3 - Desired tip acceleration profile.

The DELIGHT.MaryLin environment is then entered and the system is initialized (using "solve"). One may then run as many iterations as are desired. It is possible to interact with the process, observing eigenvalues and using the Pcomb to see how the optimization is progressing.

Included on the following pages of this chapter are the DELIGHT.MaryLin Files, the Iteration Gain History and the Starting and Final Iterations of the Assigned Variable Plots and Pcombs.

```
                       #  File mh1S    #

 Ninputs =1
 Nstates =6
 Noutputs = 2
 matrix_sizes

   # Parameters in the plant.

 R1 = 9*2*PI
 R2 = 20.6*2*PI
 R3 = 9.5*2*PI
 R4 = 0.15664
 A1 = 0.04*R2+0.22*R1
 A2 = R1*R1+0.0088*R2*R1+R2*R2
 A3 = 0.22*R2*R2*R1+0.04*R2*R1*R1
 A4 = R2*R2*R1*R1
 B2 = R1*R2*R1*R2/(R3*R3)
 B3 = 2*B2*R3
 B4 = B2*R3*R3
 C1 = (A1*A1-A2)*0.38*R4
 C2 = -A1*0.38*R4
 C3 = 0.38*R4
 C4 = B2*0.38*R4

   # System matrices

readmatrix ~expressions A
-A1          1          0          0          0          0
-A2          0          1          0          B2         0
-A3          0          0          1          B3         0
-A4          0          0          0          B4         0
0            0          0          0          0          1
0            0          0          0          0          -7.29

readmatrix  B
0
0
0
0
0
15.33

readmatrix ~expressions C
R4           0          0          0          0          0
C1           C2         C3         0          C4         0


readmatrix UFORCE
0.3

   # Design Parameters

sim_design_parameter K1_1   variation=10
sim_design_parameter K1_2   variation=0.1
sim_design_parameter K1_3   variation=0.001
sim_design_parameter K1_4   variation=0.00001
sim_design_parameter K1_5   variation=50
sim_design_parameter K1_6   variation=2.5

  #  Simulation Outputs

simulation_output Ytr(1)
simulation_output Ytr(2)
simulation_output Xtr(5)

  #  Parameters used by algorithm

Nparam = 6
Nfineq = 2
Nfmulticost = 2

step_bounds Y       PcOvershoot = 5%                 \
                    PcRise = 95%        Trise=0.8     \
                    PcSettle = 2%       Tsettle=1.8  \
                    FinalValue = 0.3        Tfinal=5
```

```
                          #  File mh1P    #

set K1_1 = 10.01176
set K1_2 = -0.1517682
set K1_3 = -0.001351515
set K1_4 = 0.00002304485
set K1_5 = 67.1955
set K1_6 = 2.522692
```

```
                         # file mh1FI  #

prob_function fineq


 constraint 1 'overshootU' Yout <= good=Y_Upper bad=1.10*Y_Upper soft using {
        Yout=Ytr(1)/0.15664
        }
     for_every TIME from 0 to 5.0 initially by 0.2

 constraint 2 'overshootL' Yout >= good=Y_Lower bad=.90*Y_Lower soft  using {
        Yout=Ytr(1)/0.15664
        }
     for_every TIME from 0.8 to 4.8 initially by 0.2

end_fineq
```

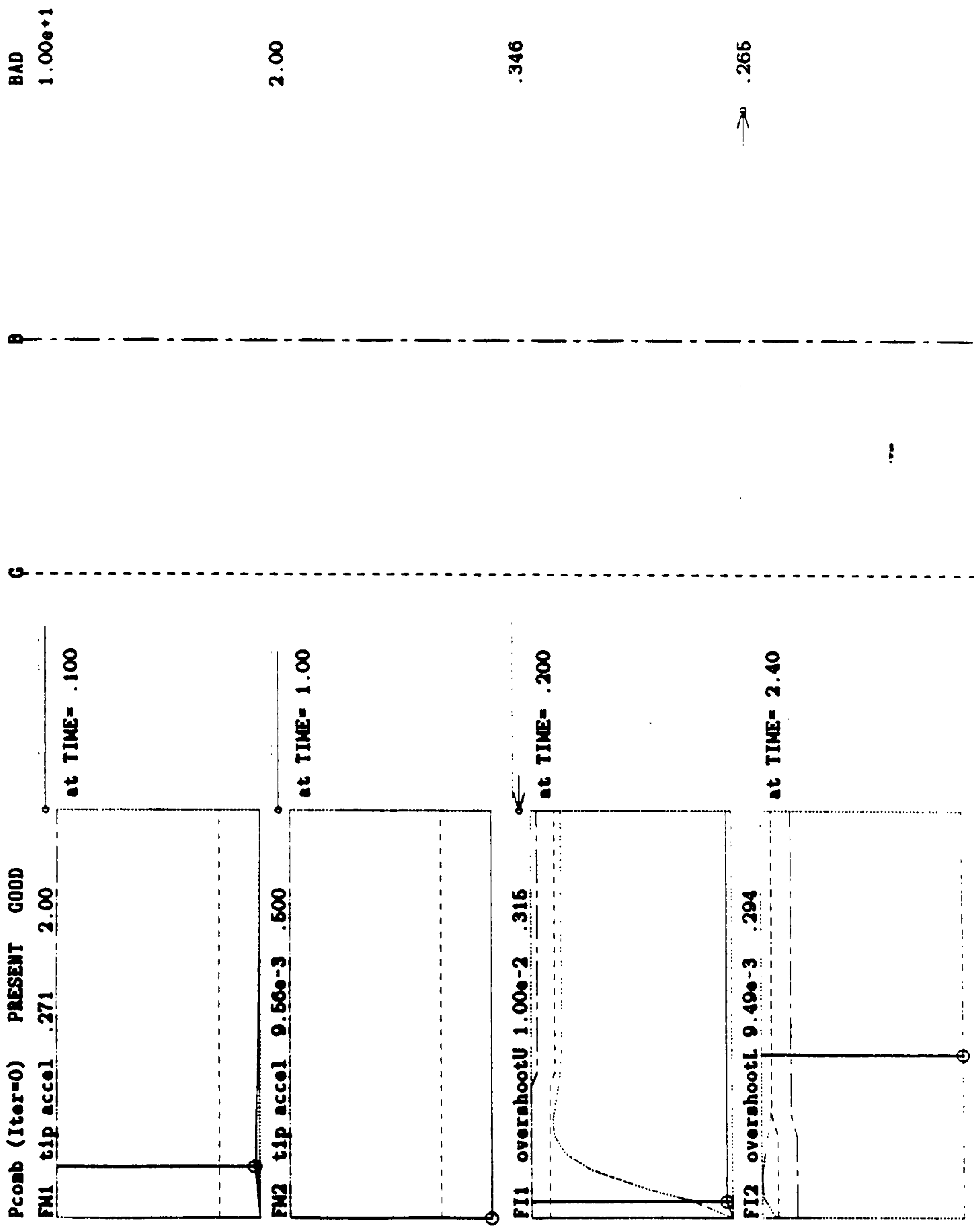
```
                         # file mh1FM  #

prob_function fmulticost


     objective 1  'tip accel slew' minimize abs(Ytr(2)) good=2 bad=10
     for_every TIME from 0 to 0.8 initially by 0.1

     objective 2  'tip accel slew' minimize abs(Ytr(2)) good=0.5 bad=2
     for_every TIME from 1.0 to 5 initially by 0.2


end_fmulticost
```

```
Parameter  Value       %wrt 0 Prev Iter=1 Phase2 MxM+SC=8.231
 1 K1_1    9.079       -9%  -9%
 2 K1_2   -.1534        1%   1%
 3 K1_3   -1.398e-3     3%   3%
 4 K1_4    2.451e-5     6%   6%
 5 K1_5    4.388e+1    -35% -35%
 6 K1_6    2.523        0%   0%

Parameter  Value       %wrt 0 Prev Iter=2 Phase2 MxM+SC=7.242
 1 K1_1    8.958       -11%  -1%
 2 K1_2   -.1536        1%   0%
 3 K1_3   -1.405e-3     4%   0%
 4 K1_4    2.469e-5     7%   1%
 5 K1_5    4.098e+1    -39%  -7%
 6 K1_6    2.512        0%   0%

Parameter  Value       %wrt 0 Prev Iter=3 Phase2 MxM+SC=6.883
 1 K1_1    8.927       -11%   0%
 2 K1_2   -.1537        1%   0%
 3 K1_3   -1.407e-3     4%   0%
 4 K1_4    2.474e-5     7%   0%
 5 K1_5    4.026e+1    -40%  -2%
 6 K1_6    2.508       -1%   0%

Parameter  Value       %wrt 0 Prev Iter=4 Phase2 MxM+SC=6.681
 1 K1_1    8.911       -11%   0%
 2 K1_2   -.1537        1%   0%
 3 K1_3   -1.408e-3     4%   0%
 4 K1_4    2.476e-5     7%   0%
 5 K1_5    3.990e+1    -41%  -1%
 6 K1_6    2.506       -1%   0%

Parameter  Value       %wrt 0 Prev Iter=5 Phase2 MxM+SC=4.355
 1 K1_1    8.478       -15%  -5%
 2 K1_2   -.1565        3%   2%
 3 K1_3   -1.516e-3    12%   8%
 4 K1_4    2.422e-5     5%  -2%
 5 K1_5    4.063e+1    -40%   2%
 6 K1_6    1.301       -48% -48%

Parameter  Value       %wrt 0 Prev Iter=6 Phase2 MxM+SC=.9970
 1 K1_1    8.275       -17%  -2%
 2 K1_2   -.1578        4%   1%
 3 K1_3   -1.570e-3    16%   4%
 4 K1_4    2.391e-5     4%  -1%
 5 K1_5    4.083e+1    -39%   1%
 6 K1_6    .6985       -72% -46%
```

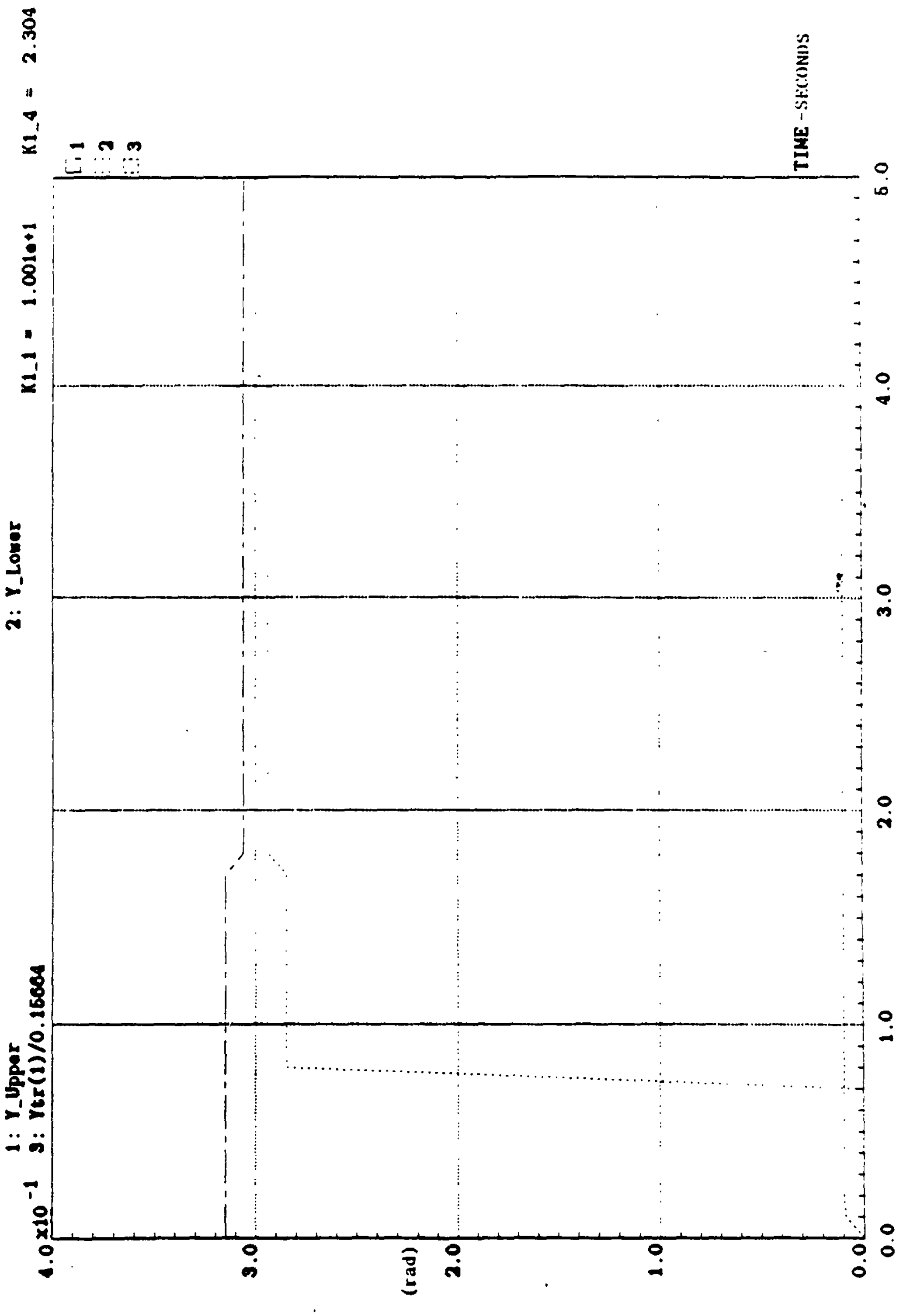
DELIGHT.MaryLin Gain History

```
Parameter   Value      %wrt 0 Prev Iter==7 Phase2 MxM+SC==.3997
 1 K1_1     8.250      -18%  0%
 2 K1_2    -.1580        4%  0%
 3 K1_3    -1.577e-3    17%  0%
 4 K1_4     2.388e-5     4%  0%
 5 K1_5     4.090e+1   -39%  0%
 6 K1_6     .6232      -75% -11%

Parameter   Value      %wrt 0 Prev Iter==8 Phase2 MxM+SC==.1634
 1 K1_1     8.248      -18%  0%
 2 K1_2    -.1580        4%  0%
 3 K1_3    -1.577e-3    17%  0%
 4 K1_4     2.388e-5     4%  0%
 5 K1_5     4.085e+1   -39%  0%
 6 K1_6     .6229      -75%  0%

Parameter   Value      %wrt 0 Prev Iter==9 Phase2 MxM+SC==2.735e-2
 1 K1_1     8.235      -18%  0%
 2 K1_2    -.1580        4%  0%
 3 K1_3    -1.580e-3    17%  0%
 4 K1_4     2.386e-5     4%  0%
 5 K1_5     4.090e+1   -39%  0%
 6 K1_6     .5853      -77% -6%
```

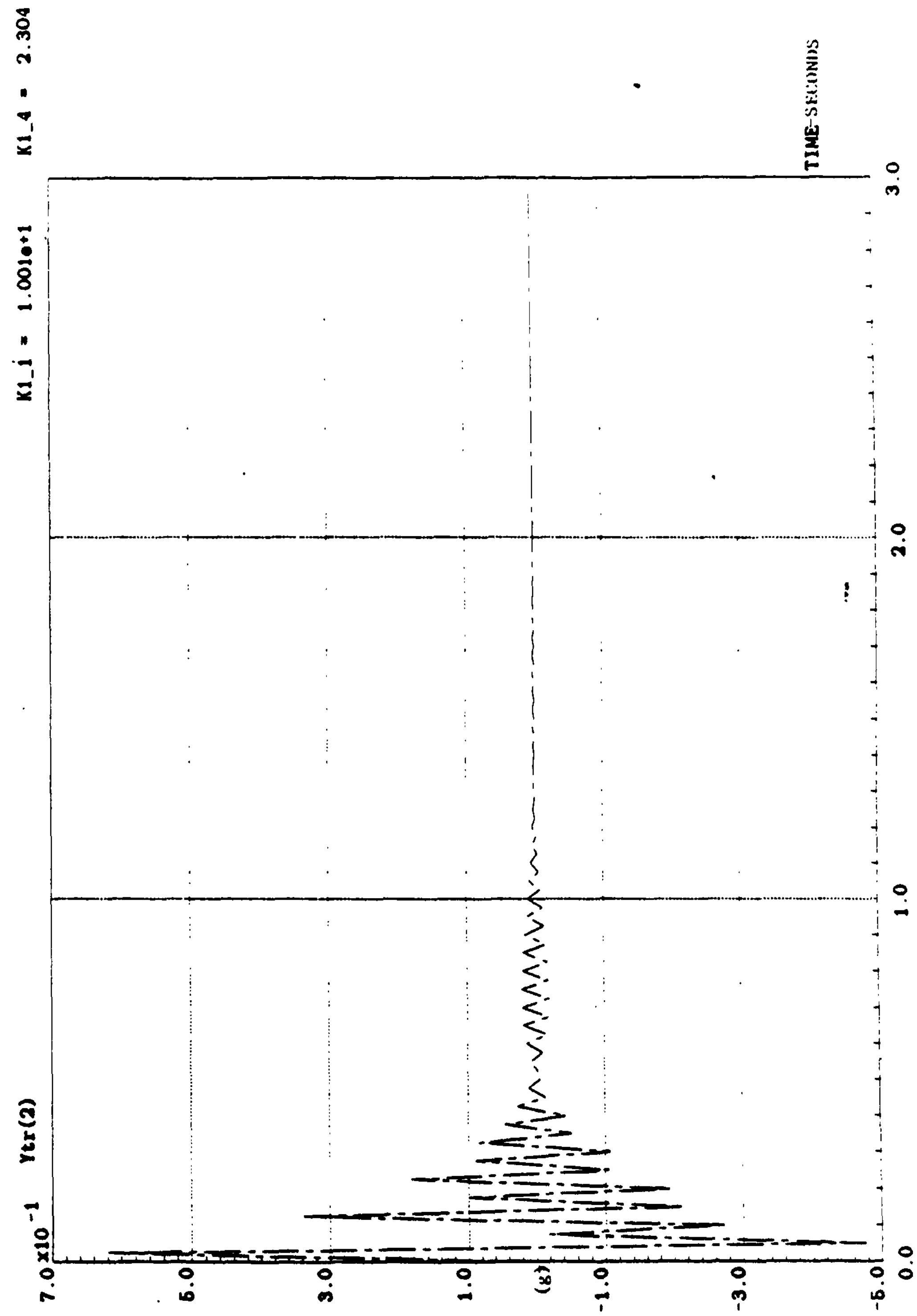
DELIGHT.MaryLin Gain History (Continued)



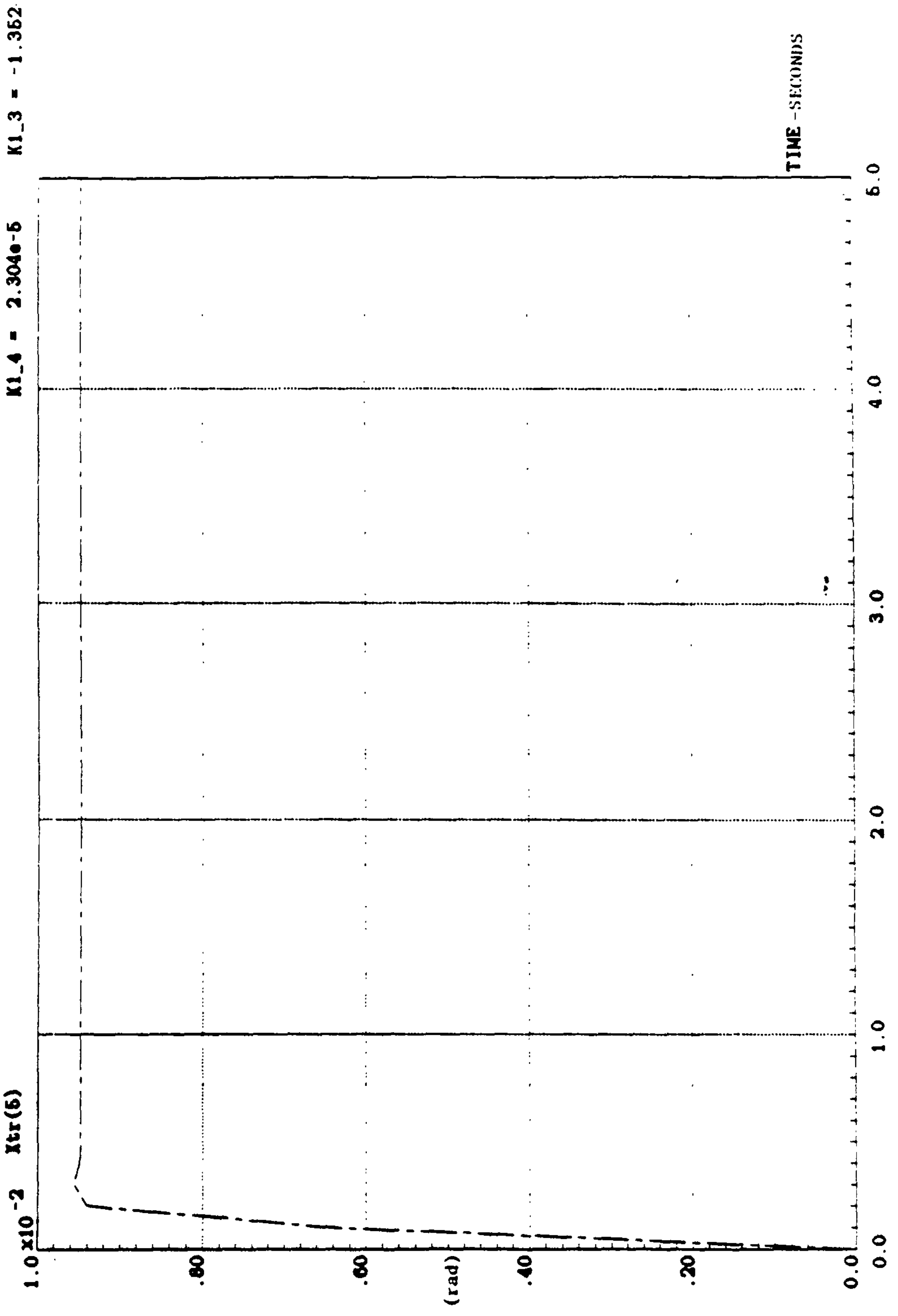
Pcomb - Starting Iteration



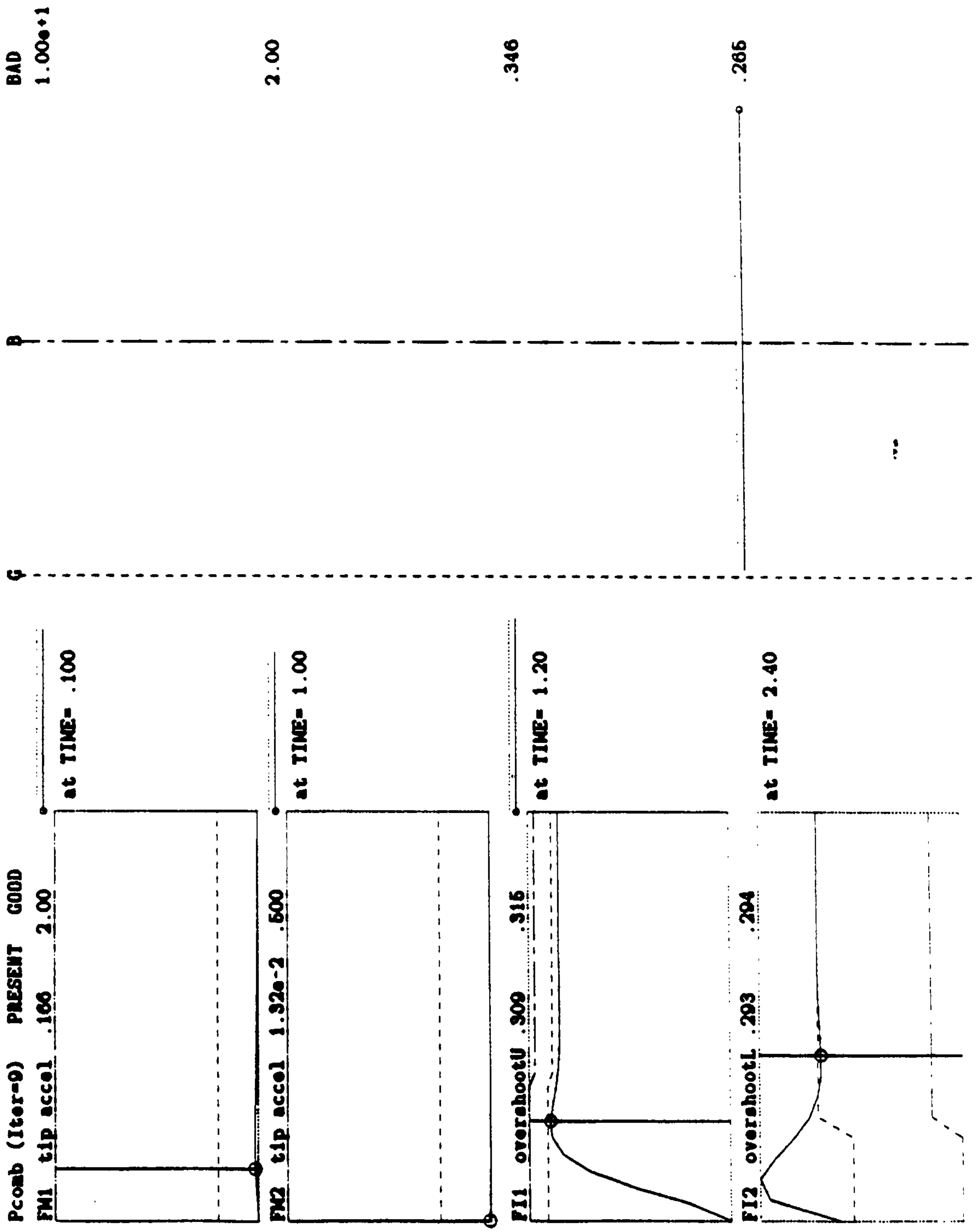
Tip Position - Starting Iteration



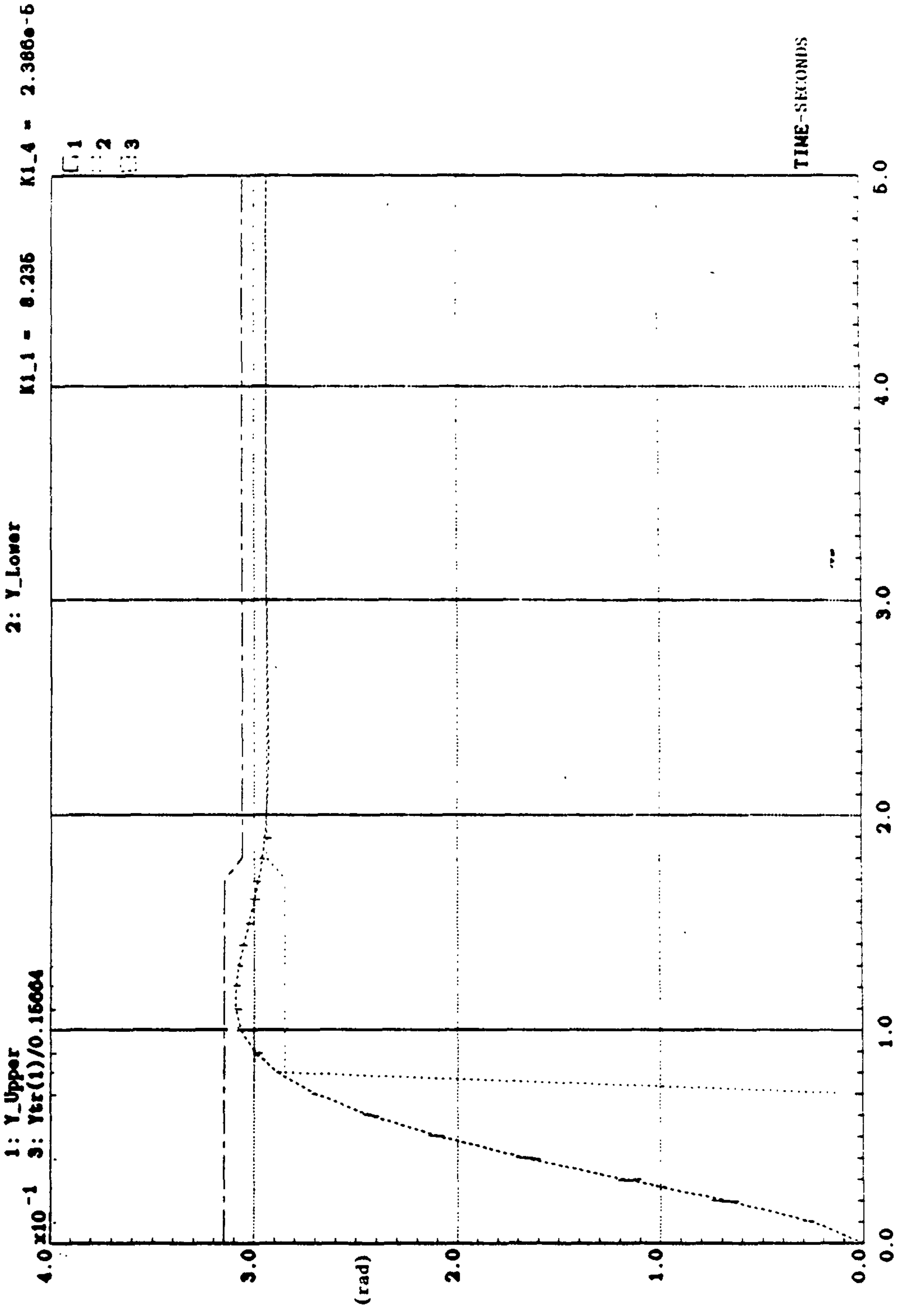
Tip Acceleration - Starting Iteration



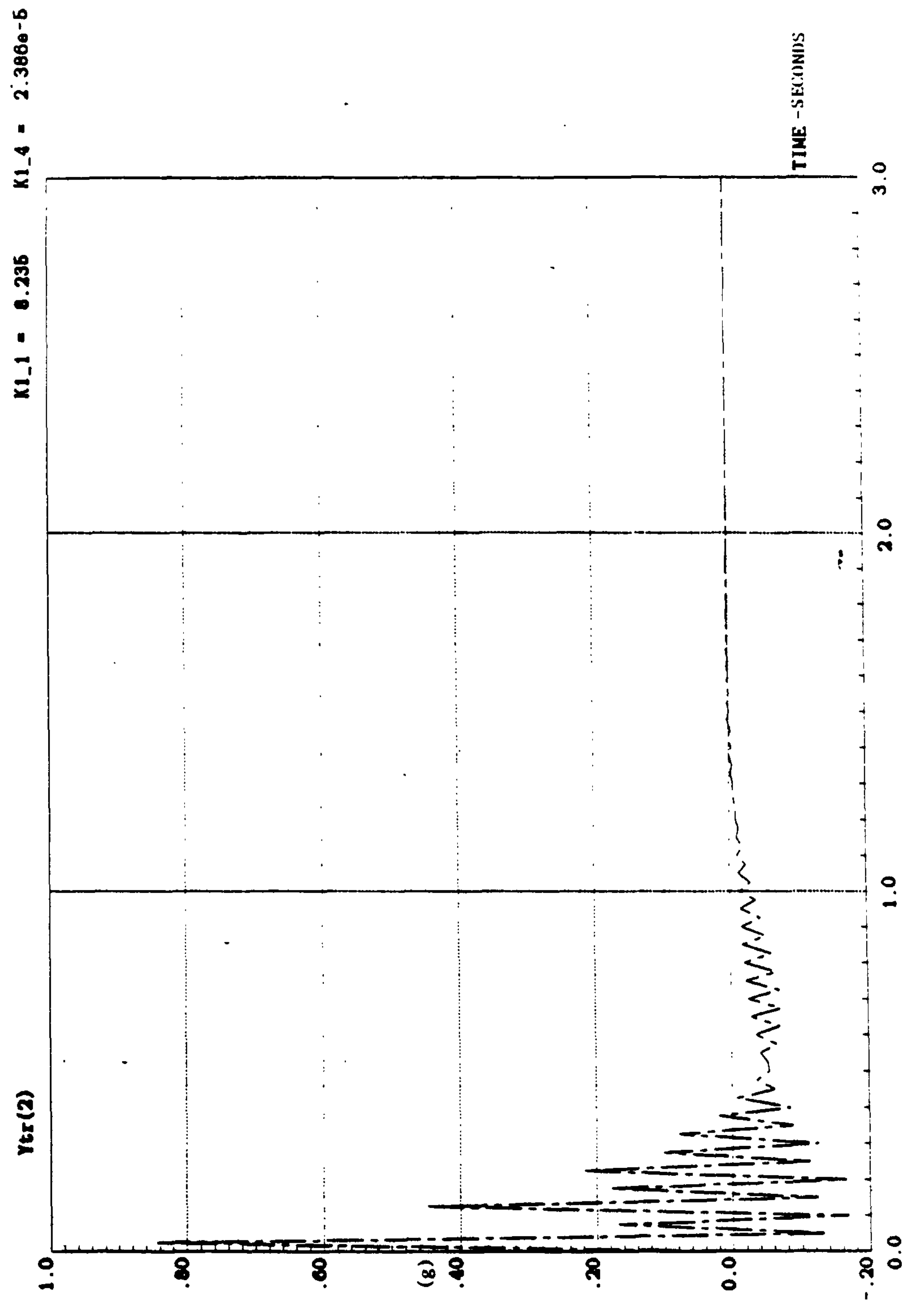
Shaft Position - Starting Iteration



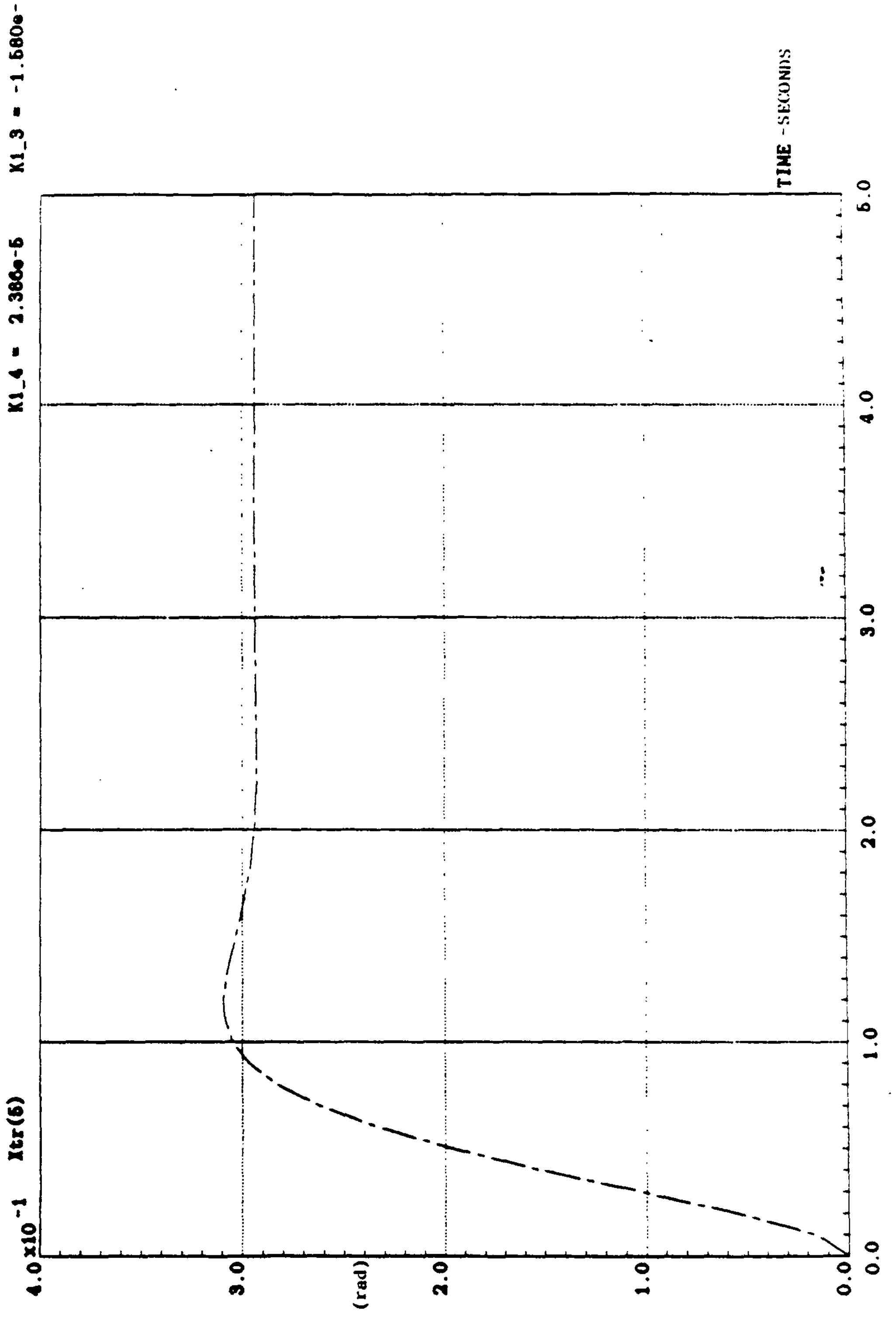
Pcomb - Final Iteration



Tip Position - Final Iteration



Tip Acceleration - Final Iteration



Shaft Position - Final Iteration

VI. OBSERVER-CONTROLLER

The pole placement design algorithm used by DELIGHT.MaryLin assumes that all of the plant states are available to drive the feedback gain vector, K. In this experimental system, as well as many other practical systems, the state vector may be incomplete or the expense of making it complete by adding plant sensors is prohibitive. Given a scheme to augment the known plant states with mathematically correct estimates of the unknown states, the "designed" system's stability and performance may be retained.

The subject of this chapter, the Observer-Controller, is such a scheme. It utilizes the plant input, plant dynamics, and feedback from plant sensor outputs to develop a complete state vector thereby providing full state feedback. Consider the linear, time-invariant, continuous time system of Figure 1.

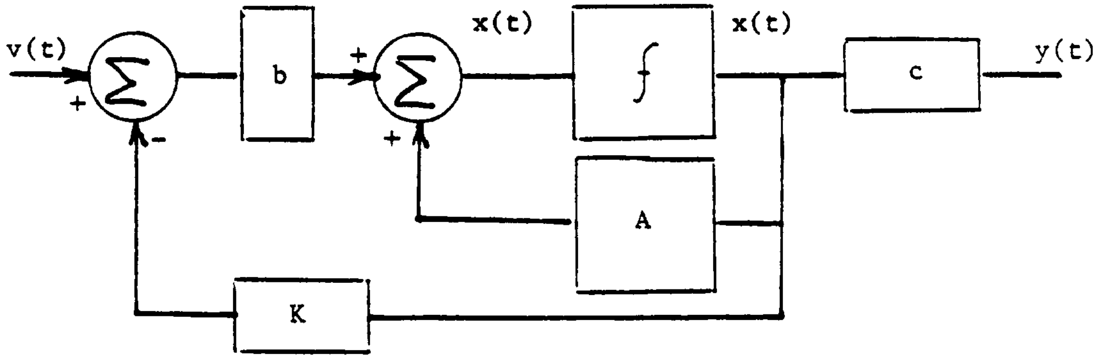


Figure 1 Linear, time-invariant system with full-state feedback

$v(t)$	- system input
$u(t)$	- error signal $(v-Kx)$
$y(t)$	- system output
A,b,c	- system matrices
K	- DELIGHT.MaryLin optimized gain

The eigenvalues of the system (A,b,c) may be arbitrarily relocated by state feedback gain K if the realization (A,b,c) is completely controllable, that is, if

$$\mathcal{C} = [b\ Ab \cdot \ \cdot \ \cdot\ A^{n-1}b] \text{ is of rank } n$$

To show this, define

$$a(s) \triangleq \det\,(sI-A)$$
$$a_K(s) \triangleq \det\,(sI-A+bK) \tag{1}$$

where K is chosen such that $a_K(s) = \alpha(s)$, $\alpha(s)$ having the desired eigenvalues.

$$a_K(s) = \det\ (sI-A)\ [I+(sI-A)^{-1}bK]$$
$$a_K(s) = \underbrace{\det(sI-A)}_{\text{our original system, } a(s)} \det\,[I+(sI-A)^{-1}bK]$$
$$a_K(s) = a(s) \det\,[I+(sI-A)^{-1}bK]$$
$$a_K(s) - a(s) = a(s)\ K(sI-A)^{-1}b$$

Recall that

$$\det(sI-A) = s^{n}+a_1s^{n-1}+a_2s^{n-2}+\cdot\cdot\cdot$$

and

$$(sI-A)^{-1} = 1/s[I-A/s]^{-1} = [I/s + A/s^2 + A^2/s^3 + \cdots]$$

and

$$a(s)(sI-A)^{-1} = \frac{[I/s + A/s^2 + A^2/s^3 \cdots]}{[s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots]}$$

$$= s^{n-1}I + s^{n-2}(A+a_1 I) + s^{n-3}(A^2+a_1 A+a_2 I) + \cdots$$

Equating like powers of s on both sides

$$\begin{aligned}
\alpha_1 - a_1 &= Kb \\
\alpha_2 - a_2 &= K(A+a_1 I)b \\
\alpha_3 - a_3 &= K(A^2+a_1 A+a_2 I)b \\
\cdot \quad &\quad \cdot \\
\cdot \quad &\quad \cdot \\
\cdot \quad &\quad \cdot
\end{aligned}$$

this result may be arranged using the controllability matrix, $\mathcal{C}$

$$[\alpha - a] = K\,[b \;\; Ab \;\; A^2 b \;\cdots\;] \begin{bmatrix} 1 & a_1 & a_2 & a_3 & \cdots \\ & 1 & a_1 & a_2 & \cdots \\ & & 1 & a_1 & \cdots \\ & & & 1 & \cdots \\ & \bigcirc & & \cdot & \\ & & & \cdot & \end{bmatrix}$$

obviously nonsingular

Then if R__ is defined as a lower triangular Toeplitz matrix with first column $[1 \; a_1 \; a_2 \; \cdot \; \cdot \; \cdot \;]'$

$$[\alpha \; -a] = K \mathcal{C} \; R\_'$$

solving for K

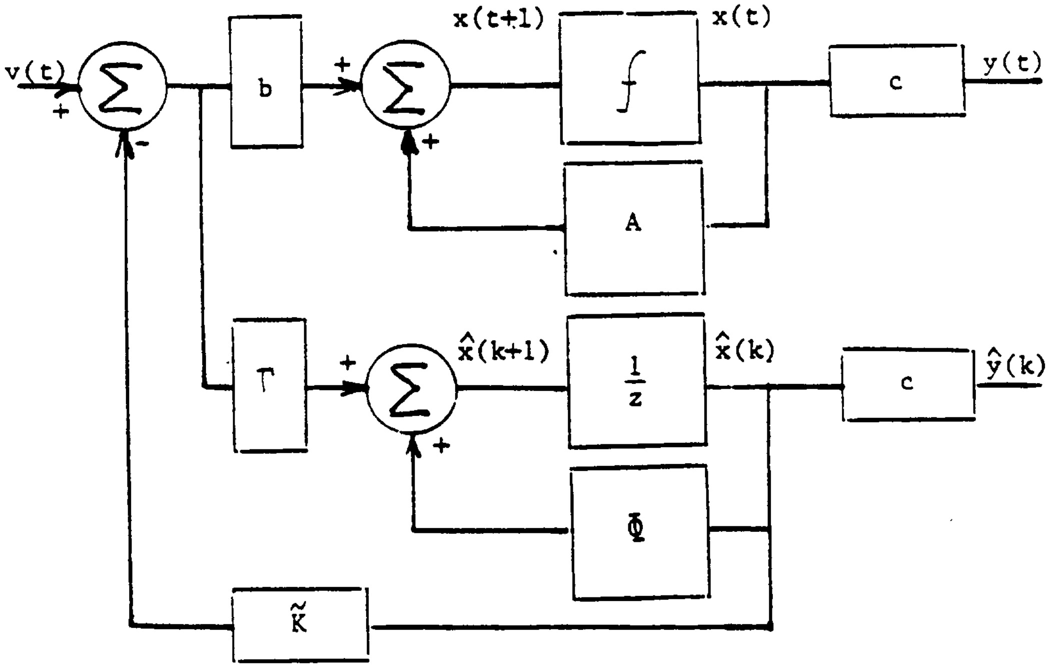
$$K = [\alpha \; -a](R\_')^{-1} \mathcal{C}^{-1}$$

from which it is seen that to be able to arbitrarily specify $[\alpha \; -a]$ one must have $\mathcal{C}$ full rank.

While it will not always be necessary to have full state feedback to provide eigenvalue relocation, arbitrary relocation will usually require that full state feedback be available.

The system of Figure 1 includes the plant realization (A,b,c) and also the DELIGHT.MaryLin derived feedback gain vector, K. This system, given that the plant develops a complete state vector, has the same dynamics as optimized in DELIGHT.MaryLin via system engineering specifications. In our physical plant, only two of the six states are measurable (x_6, shaft rate and x_5, shaft position) along with tip acceleration, $\ddot{x}_1$ (which may also be expressed as a combination of x_1, x_2, x_3 and x_5). One must now develop estimates for the four states not measurable in the plant.

Returning to the Observer-Controller, Figure 1 is enhanced with a model of the plant which is dynamically equivalent to the plant. This plant model is the (A,b,c) realization obtained from the modal identification experiment, but since this model is implemented in real time software on the PC, the system is translated to discrete time via a six term matrix exponential series expansion. When the model is running in software, all of the six states may be measured and may be used as estimates of the plant states. This open loop observer-controller is shown in Figure 2.



v(t) - System input

$\dot{u}(t)$ - System error (v-Kx)

x(t) - Plant state vector

$\hat{x}(k)$ - Estimated state vector

y(t) - Plant output

$\hat{y}(k)$ - Estimated output

$\tilde{K}$ - Discretized feedback gain

Figure 2 Open Loop Observer-Controller

For an open loop observer-controller with model dynamics identical to plant dynamics, initial conditions in the plant and the model the same, and no external disturbances, $\hat{x}$ approaches x and the observer will mimic the plant providing the necessary state feedback for control. Intuition tells us that, in the presence of disturbances, this too much to hope for and the open loop observer is modified to use, as feedback, the measured plant outputs. Define the error in the estimate to be

$$\tilde{x}(k) = \hat{x}(k)-x(k)$$

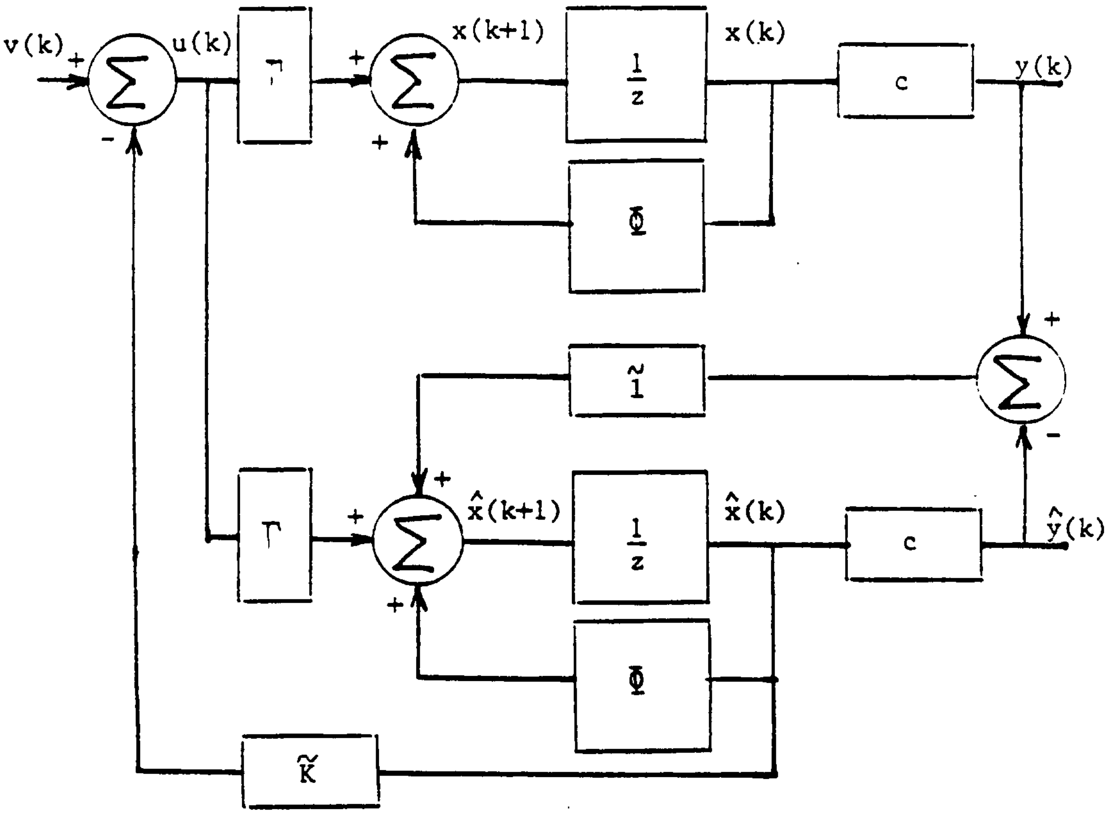
then the system representing error dynamics is

$$\tilde{x}(k+1) = A\tilde{x}(k)$$

which again depends on the dynamics of A (or Φ in the discrete time system). Intuition also tells us that the error can be driven asymptotically to zero by appropriately chosen feedback. Figure 1 is further enhanced to obtain Figure 3, the closed loop Observer-Controller. Here the plant outputs are sampled to correct the model inputs, and so the plant must be translated to discrete time notation. Now the error dynamics depend on $(\Phi-\tilde{l}c)$, that is,

$$\tilde{x}(k+1) = [\Phi-\tilde{l}c]\tilde{x}(k)$$

and, if the eigenvalues of $(\Phi-\tilde{l}c)$ are asymptotically stable, $\hat{x}(k)$ will approach x(k). Since there is no physical impediment (such as plant time constant) other than computer speed, the eigenvalues of $(\Phi-\tilde{l}c)$ can be adjusted to be considerably faster than the eigenvalues of the plant. To show that one needs only an observable system in order to



$\tilde{l}$ - discretized Observer feedback gain

Figure 3 Closed loop Observer-Controller

arbitrarily place eigenvalues of the closed loop observer by appropriate choice of l, refer to Equation 1 on page 47 and define

$$a_1(z) = \det\,(zI-\Phi-\tilde{l}c)$$

Following a similar derivation,

$$a_1(z)-a(z) = a(z)\;c(zI-\Phi)^{-1}\tilde{l}$$

Using the transpose of the observability matrix, $\mathcal{O}'$

$$[\alpha - a]' = \sigma' \underset{R_-'}{\begin{bmatrix} 1 & a_1 & a_2 & a_3 \cdots \\ & 1 & a_1 & a_2 \cdots \\ & & 1 & a_1 \cdots \\ & & & 1 \cdots \\ & \bigcirc & & \cdot \\ & & & \cdot \end{bmatrix}} \tilde{l}$$

Then

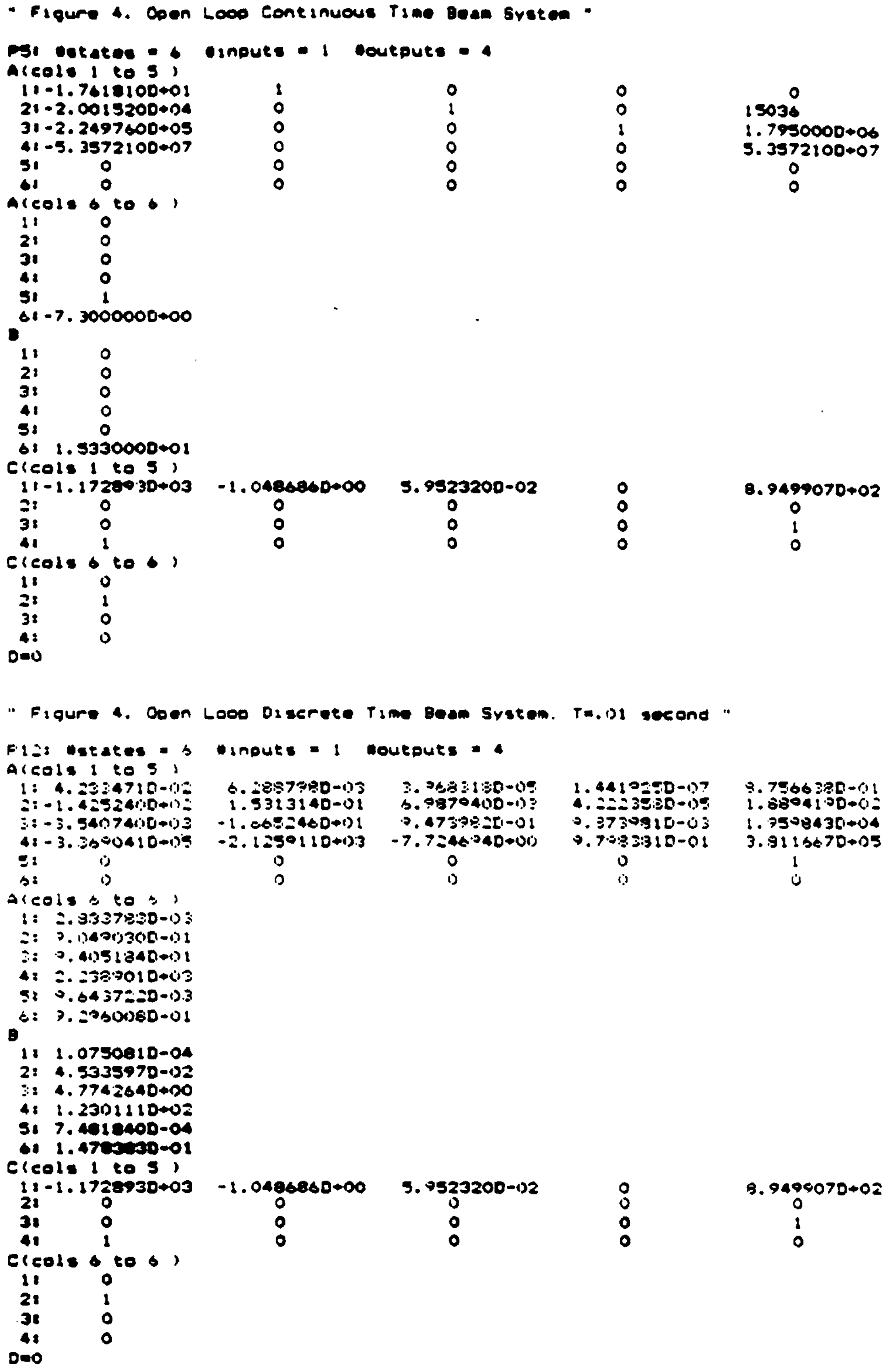
$$[\alpha - a]' = \sigma' R_-' \tilde{l}$$

solving for $\tilde{l}$

$$\tilde{l} = (\sigma')^{-1} (R_-')^{-1} [\alpha - a]'$$

and to arbitrarily specify $[\alpha - a]$, σ must be full rank. The necessary tools to implement the observer-controller now exist.

Reconsider the open loop continuous time realization of the plant shown with the discrete time realization in Figure 4. Translation was made by use of the Classical Control (CC) Pade Matrix Exponential and the resulting system was simulated to evaluate the translation. The c matrix has been expanded to develop four outputs; y_1 = tip acceleration, y_2 = shaft rate, y_3 = shaft position, and y_4 = tip position.



```
" Figure 4. Open Loop Continuous Time Beam System "

P5: #states = 6  #inputs = 1  #outputs = 4
A(cols 1 to 5 )
 1:-1.761810D+01        1              0              0              0
 2:-2.001520D+04        0              1              0          15036
 3:-2.249760D+05        0              0              1          1.795000D+06
 4:-5.357210D+07        0              0              0          5.357210D+07
 5:      0              0              0              0              0
 6:      0              0              0              0              0
A(cols 6 to 6 )
 1:      0
 2:      0
 3:      0
 4:      0
 5:      1
 6:-7.300000D+00
B
 1:      0
 2:      0
 3:      0
 4:      0
 5:      0
 6: 1.533000D+01
C(cols 1 to 5 )
 1:-1.172893D+03  -1.048686D+00  5.952320D-02        0          8.949907D+02
 2:      0              0              0              0              0
 3:      0              0              0              0              1
 4:      1              0              0              0              0
C(cols 6 to 6 )
 1:      0
 2:      1
 3:      0
 4:      0
D=0


" Figure 4. Open Loop Discrete Time Beam System. T=.01 second "

P12: #states = 6  #inputs = 1  #outputs = 4
A(cols 1 to 5 )
 1: 4.233471D-02   6.288798D-03   3.968318D-05   1.441925D-07   9.756638D-01
 2:-1.425240D+02   1.531314D-01   6.987940D-03   4.222358D-05   1.889419D+02
 3:-3.540740D+03  -1.665246D+01   9.473982D-01   9.373981D-03   1.959843D+04
 4:-3.369041D+05  -2.125911D+03  -7.724694D+00   9.798331D-01   3.811667D+05
 5:      0              0              0              0              1
 6:      0              0              0              0              0
A(cols 6 to 6 )
 1: 2.833783D-03
 2: 9.049030D-01
 3: 9.405184D+01
 4: 2.238901D+03
 5: 9.643722D-03
 6: 9.296006D-01
B
 1: 1.075081D-04
 2: 4.533597D-02
 3: 4.774264D+00
 4: 1.230111D+02
 5: 7.481840D-04
 6: 1.478383D-01
C(cols 1 to 5 )
 1:-1.172893D+03  -1.048686D+00  5.952320D-02        0          8.949907D+02
 2:      0              0              0              0              0
 3:      0              0              0              0              1
 4:      1              0              0              0              0
C(cols 6 to 6 )
 1:      0
 2:      1
 3:      0
 4:      0
D=0
```

Figure 4 (A,b,c) and (Φ, Γ ,c) Realization of the Plant

```
" Figure 5, DELIGHT.MaryLin Optimized Feedback Gain, K "

P40: #states = 0  #inputs = 6  #outputs = 1
(cols 1 to 5 )
 1: 8.235000D+00  -1.580000D-01  -1.580000D-03   2.386000D-05   4.090000D+01
(cols 6 to 6 )
 1: 5.803000D-01


" Figure 5, K-tilde, Modified Feedback Gain for Discrete Time "

P46: #states = 0  #inputs = 6  #outputs = 1
(cols 1 to 5 )
 1: 1.834145D+01  -1.097972D-01  -2.299722D-03   1.489871D-05   1.941298D+01
(cols 6 to 6 )
 1: 7.378073D-01


" Figure 5, Eigenvalues of Closed Loop Observer-Controller (Phi-Gamma*Ktilde) "

P102: #states = 0  #inputs = 6  #outputs = 6
(cols 1 to 5 )
 1: 2.569763D-01   9.303478D-01        0               0               0
 2:-9.303478D-01   2.569763D-01        0               0               0
 3:      0              0        9.752126D-01   2.633534D-02        0
 4:      0              0       -2.633534D-02   9.752126D-01        0
 5:      0              0              0               0        7.382387D-01
 6:      0              0              0               0       -5.049293D-01
(cols 6 to 6 )
 1:      0
 2:      0
 3:      0
 4:      0
 5: 5.049293D-01
 6: 7.382387D-01
```

Figure 5   Continuous and Discrete Time Controller Gains

Figure 5 is a listing of the DELIGHT.MaryLin optimized gain, K and the discretized gain, $\tilde{K}$. The discrete gain is obtained by comparison of characteristic equations of:

1. An open loop discrete time system to which feedback is added and
2. A continuous time closed loop system translated to discrete time.

Equating like powers of h (through the second order), poles up to and including the second order (h^2) are equated. (Reference: Astrom and Wittenmark, pp. 189-192)

$$\tilde{K} = K[I+(A-bK)h/2], \quad h = .01 \text{ second}$$

The eigenvalues of the system ($\Phi - \Gamma K$) are also included in Figure 5. Note the following:

eigenvalue	radius	ζ	τ	f_n(hz)
1	.965	.027	.28	20.6
2	.975	.68	.39	.6
3	.894	.183	.09	9.0

Table 1

There is no dominant system pole, and the radii are all large, implying that the system will be quite sensitive to any gain adjustments and to non-exactness of the state vector used for feedback.

Figure 6 is a comparison of beam tip step responses for the DELIGHT.MaryLin optimized system [(A-bK),b,c] with the solid line plotted against the discrete time observer [($\Phi - \Gamma$ K), Γ ,C] denoted by the small circles. Although the system responses are essentially the same, the continuous time system settles somewhat faster (1.8 seconds to zero error) compared with the discrete time system (1.8 seconds to 1.5% error, 2.15 seconds to zero error). This testing confirms that the controller gain optimized by DELIGHT.MaryLin and digitized will provide the expected stability and performace.

Figure 7 is the discrete system's beam tip acceleration response to a step input. Note its similarity to the final iteration of the DELIGHT.MaryLin optimized (continuous time) system on page 44.

The observer output feedback gain must be developed such that the estimated state vector approaches the plant state vector asymptotically and with faster poles than those of the plant. A two step approach was used to achieve this. First, a new DELIGHT.MaryLin optimization was performed using the (A,b,c) realization developed from the voltage to shaft position modal identification transfer function. This model includes the shaft inertia and was truncated to include only the inertia of beam that is carried rigidly (i.e., the first bending mode). Here the shaft settling response may be optimized. The resulting shaft position and shaft rate gains, along with a newly introduced tip acceleration gain, were then modified by pole placement to produce asymptotic settling of the three outputs.

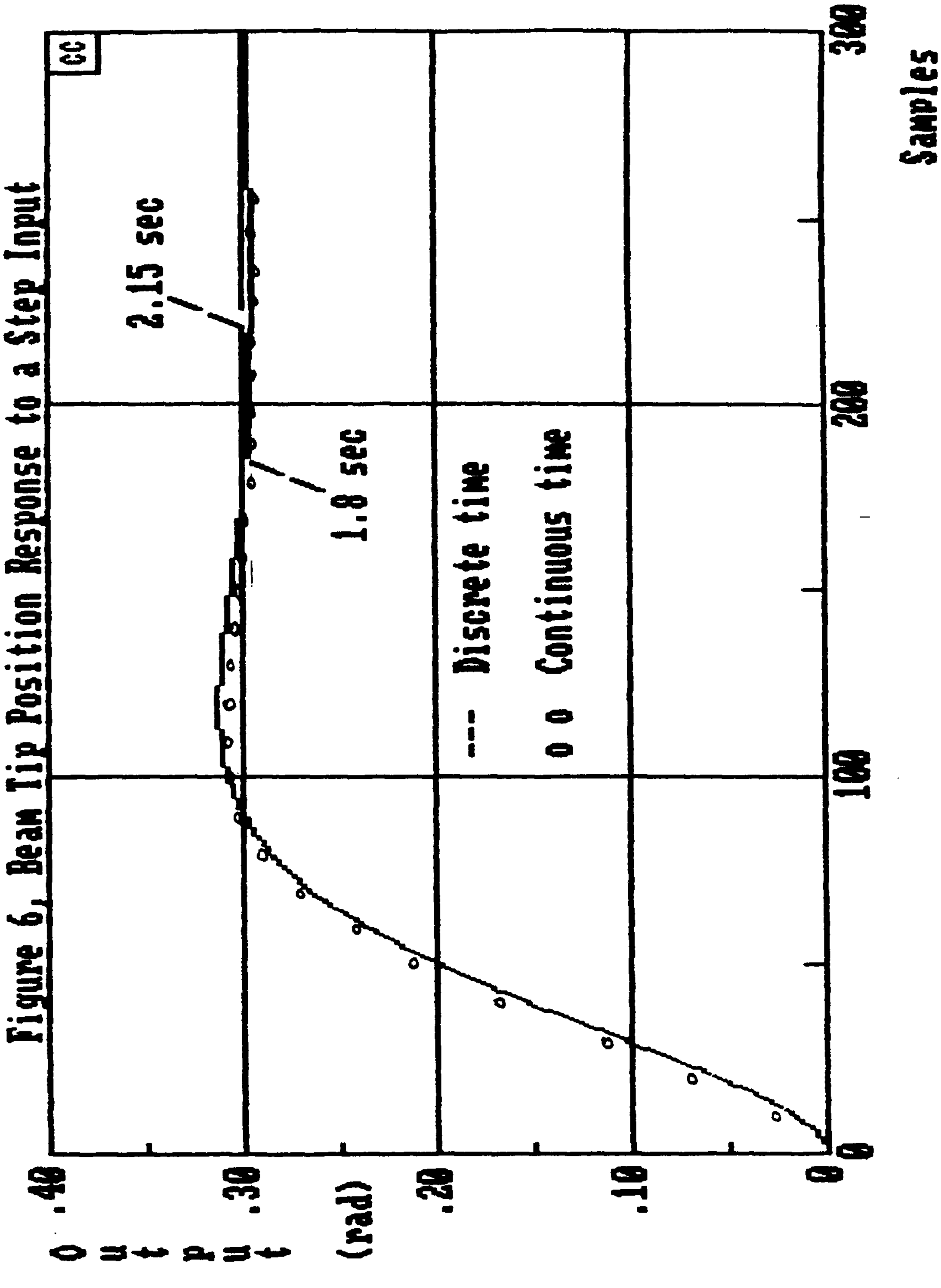


Figure 6 Comparison of Beam Tip Step Responses

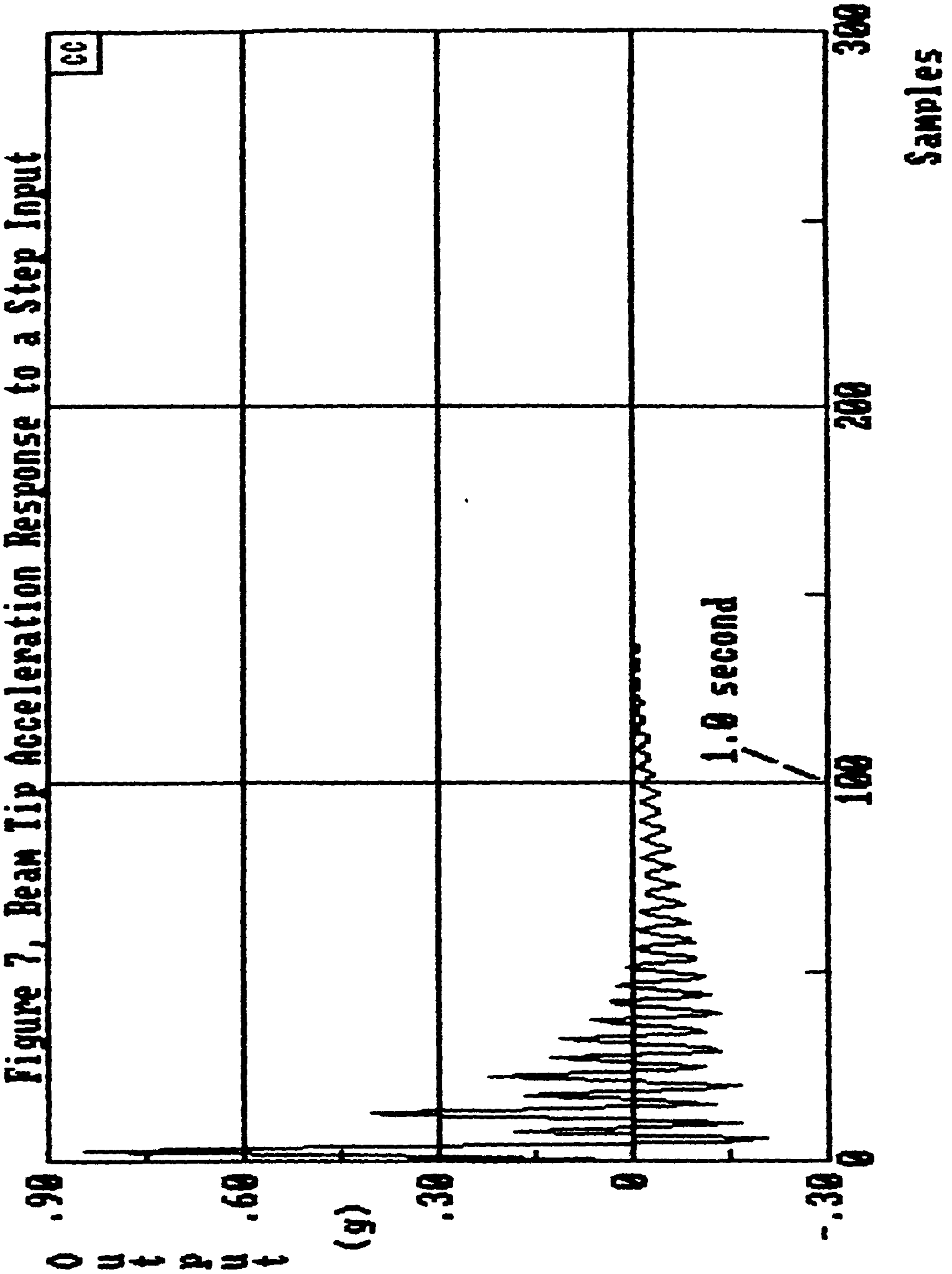


Figure 7 Observer Beam Tip Acceleration Response to a Step Input

Figure 8 shows the discrete observer output feedback gain, L, where

$$\tilde{l} = \Gamma \tilde{L}$$

This mechanization was allowed (although limiting feedback possibilities by forcing $\tilde{l}$ to have a factor, Γ) to obtain laboratory readjustment of the tip acceleration feedback gain. As was stated earlier, the dominant tip mode (20.6 hz) will have a 72° phase shift, inherent in the digitizing process, between the plant sensor and the observer. It was felt that some gain adjustment might be necessary here. Figure 8 also shows the eigenvalues of the observer with output feedback (Φ -Γ $\tilde{L}$c) or, referring to Figure 3, (Φ-$\tilde{l}$c).

```
" Figure 8. Observer Feedback Gain, L-tilde "

P6: #states = 0  #inputs = 3  #outputs = 1

 1: 1.000000D-02   7.500000D-01   1.015000D+01

" Figure 8. Eigenvalues of the Observer with Output Feedback (Phi-Gamma*Ltilde*c
) "

P29: #states = 0  #inputs = 6  #outputs = 6
(cols 1 to 5 )
 1: 2.660211D-01   9.449933D-01       0               0               0
 2:-9.449933D-01   2.660211D-01       0               0               0
 3:      0             0         7.908480D-01   4.971591D-01       0
 4:      0             0        -4.971591D-01   7.908480D-01       0
 5:      0             0              0               0         9.061454D-01
 6:      0             0              0               0        -7.834406D-02
(cols 6 to 6 )
 1:      0
 2:      0
 3:      0
 4:      0
 5: 7.834406D-02
 6: 9.061454D-01
```

Figure 8 Observer Feedback gain and Closed Loop Eigenvalues.

From Figure 8 note the following

eigenvalue	radius	ζ	τ	f_n(hz)
1	.98	.015	.49	20.6
2	.93	.13	.13	9.0
3	.91	.74	.106	2.0

Table 2

Figures 9 through 11 depict the closed loop response of the Observer with output feedback. It should be noted that the plots have not been adjusted for feedback gain. Specifically, Figure 9 settles to a value of .0287 which, multiplied by its feedback gain of 10.15, yields 0.29 which is observed in Figure 6 as the final value of the step. Figure 11 must be scaled by 0.01 which shows acceleration at .5 second of less than .0015g peak.

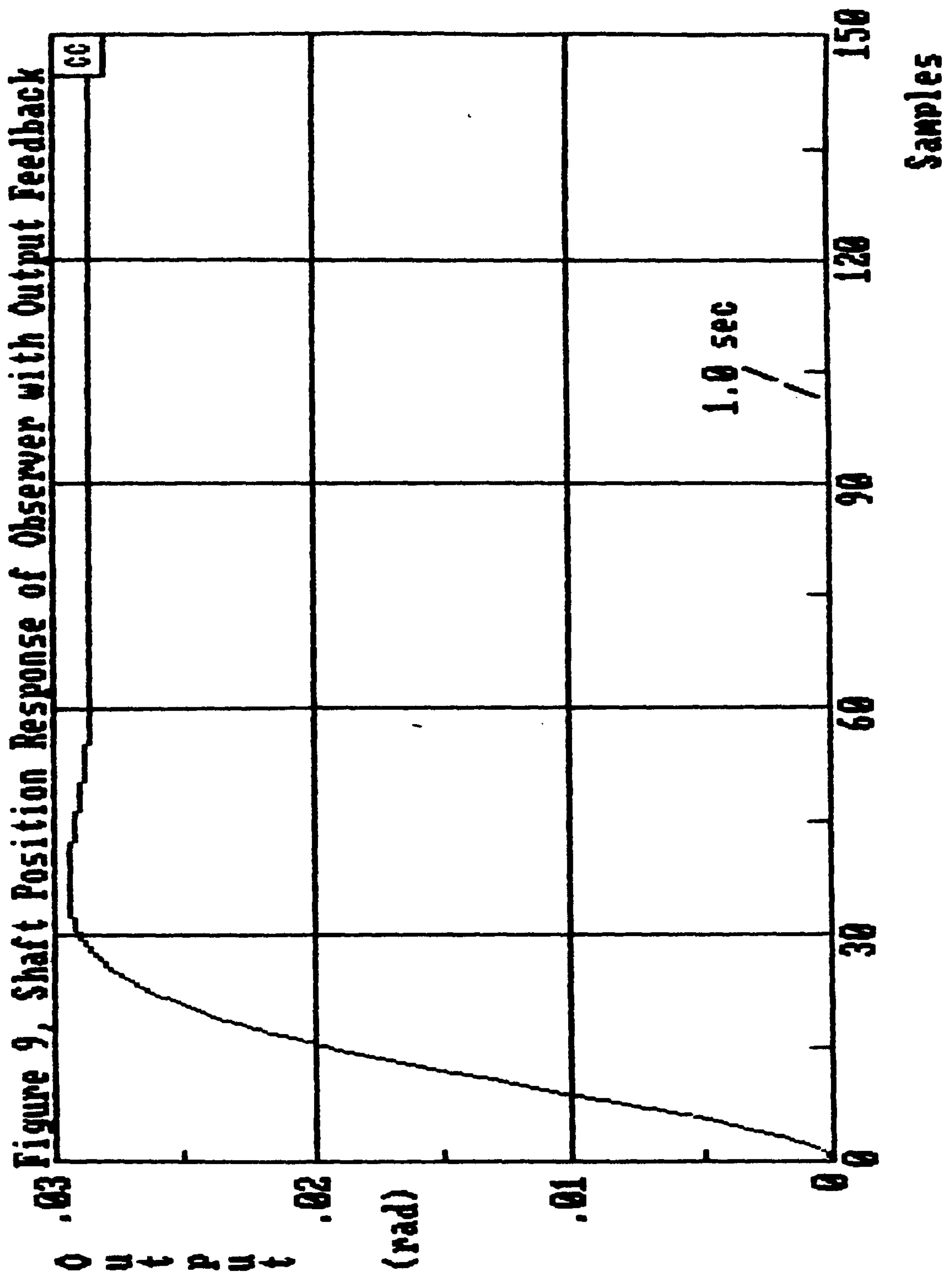


Figure 9   Shaft Position Response of Observer with Output Feedback

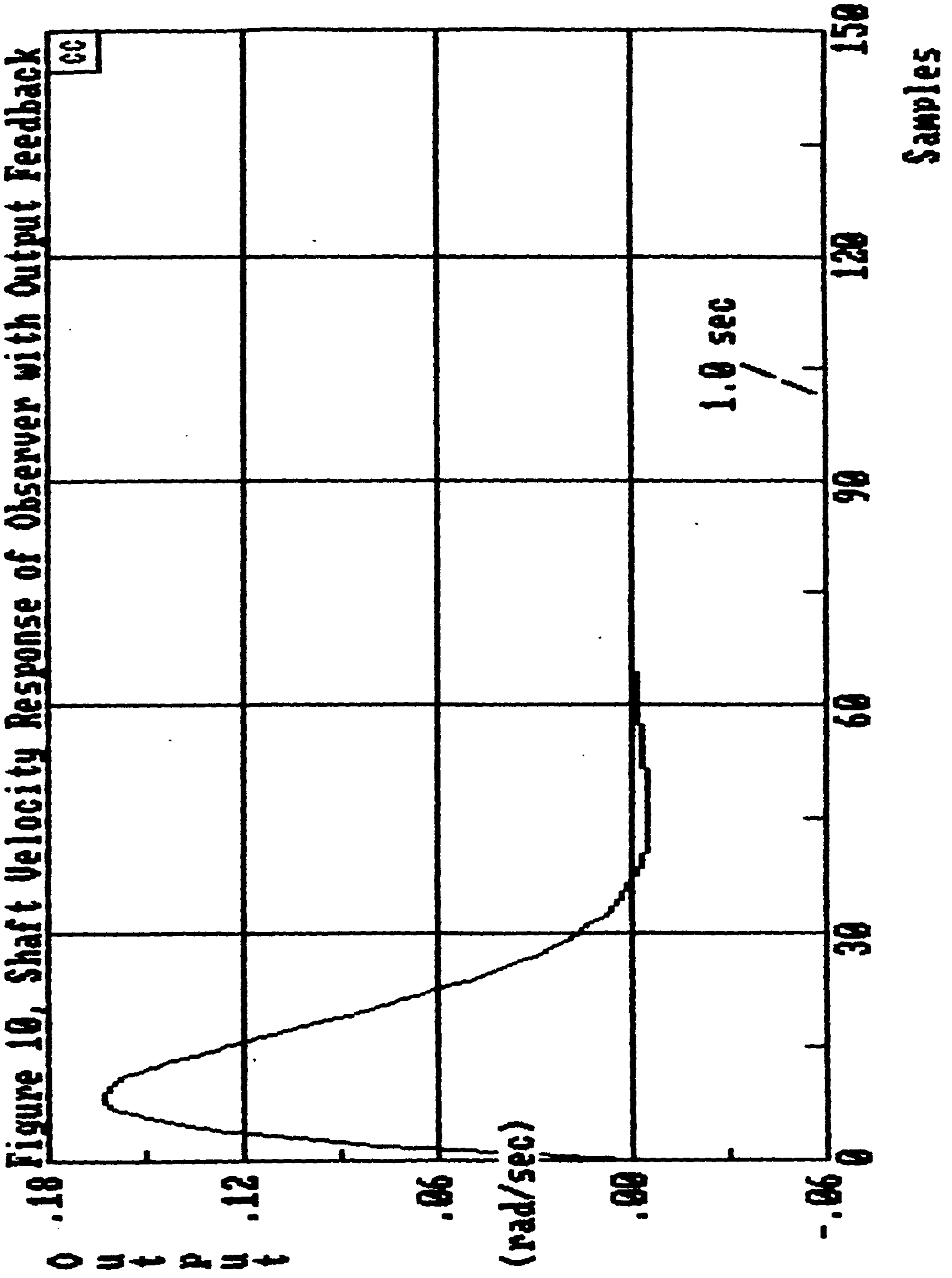


Figure 10   Shaft Rate Response of Observer with Output Feedback

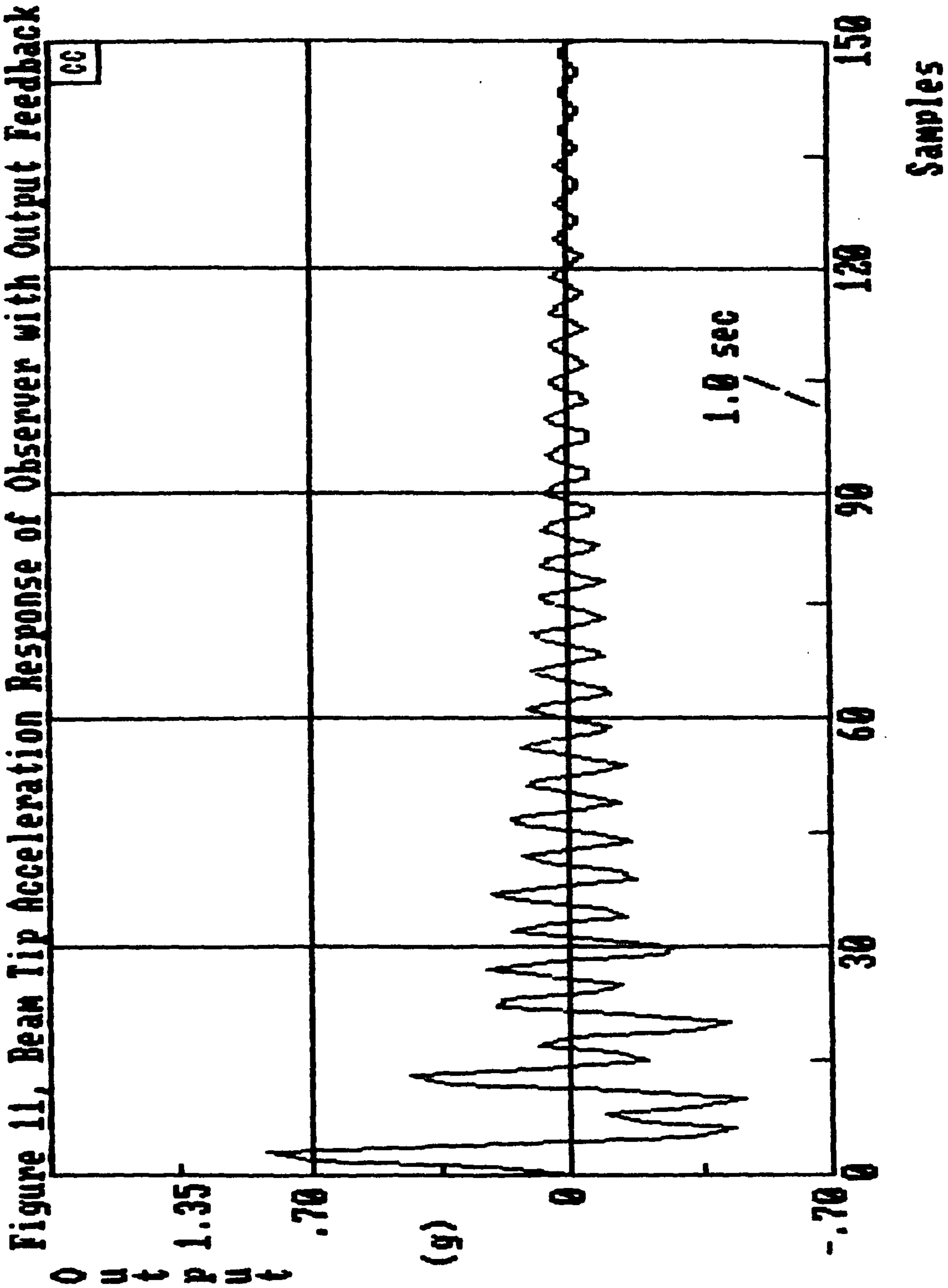


Figure 11 Tip Acceleration Response of Observer with Output Feedback

The final problem before program code is written is to determine gain scaling for the experiment. There are unavoidable gains in the system (i.e., Data Acquisiton Adapter A/D and D/A and internal integer arithmetic) which must be compensated. Referring to Figure 3 of this chapter,

$$x(k+1) = v(k) + \Phi x(k) - \Gamma K\hat{x}(k) \qquad (2)$$

for the plant, and for the observer

$$\hat{x}(k+1) = \Gamma v(k) + \Phi\hat{x}(k) - \Gamma \tilde{K}\hat{x}(k) - \tilde{l}c\hat{x}(k) + \tilde{l}cx(k)$$

$$= \Gamma v(k) + \Phi\hat{x}(k) - \Gamma \tilde{K}\hat{x}(k) - \Gamma \tilde{L}c\hat{x}(k) + \Gamma \tilde{L}cx(k) \qquad (3)$$

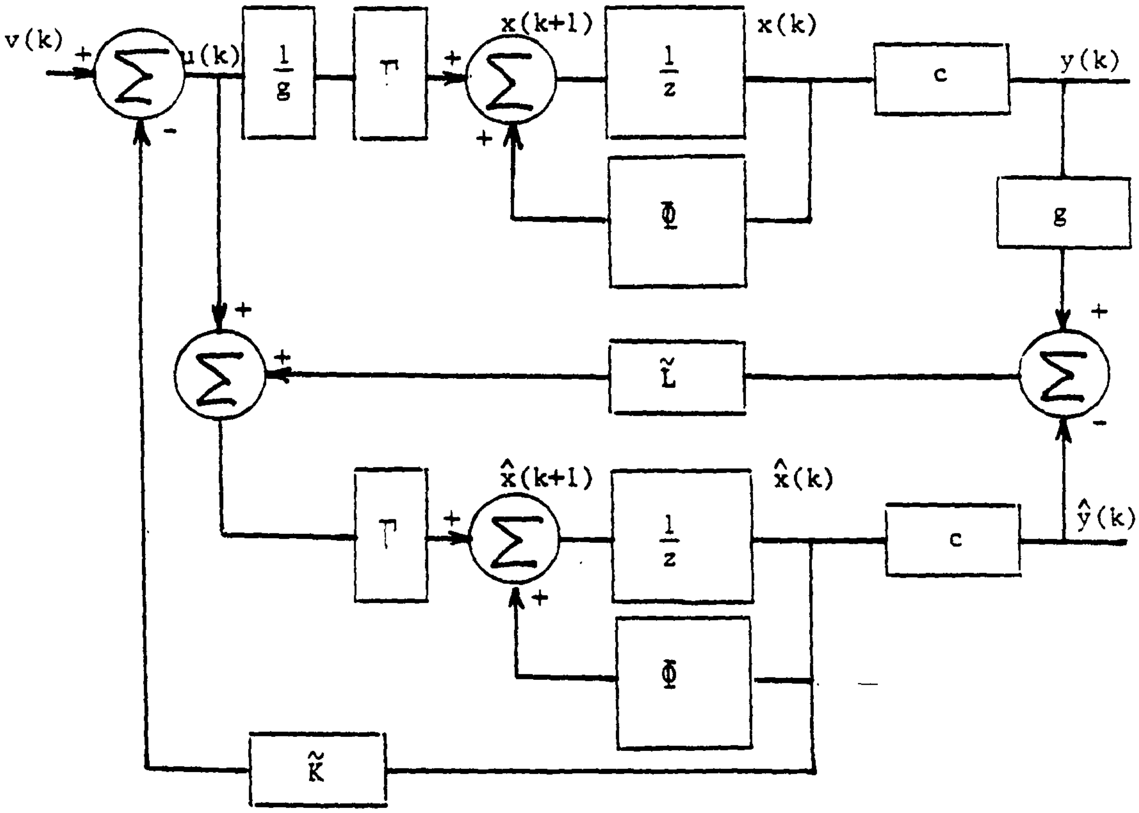
The unavoidable scaling of our experiment due to connections between the plant and the data acquisition adapter of the PC, Figure 12, is now developed. Following the same procedure,

$$x(k+1) = 1/g\,\Gamma v(k) + \Phi x(k) - 1/g\,\Gamma \tilde{K}\hat{x}(k) \qquad (4)$$

for the plant, and for the observer

$$x(k+1) = 1/g\,\Gamma v(k) + \Phi\hat{x}(k) - \Gamma \tilde{K}\hat{x}(k) - \Gamma \tilde{L}c\hat{x}(k) + \Gamma \tilde{L}cx(k) \qquad (5)$$

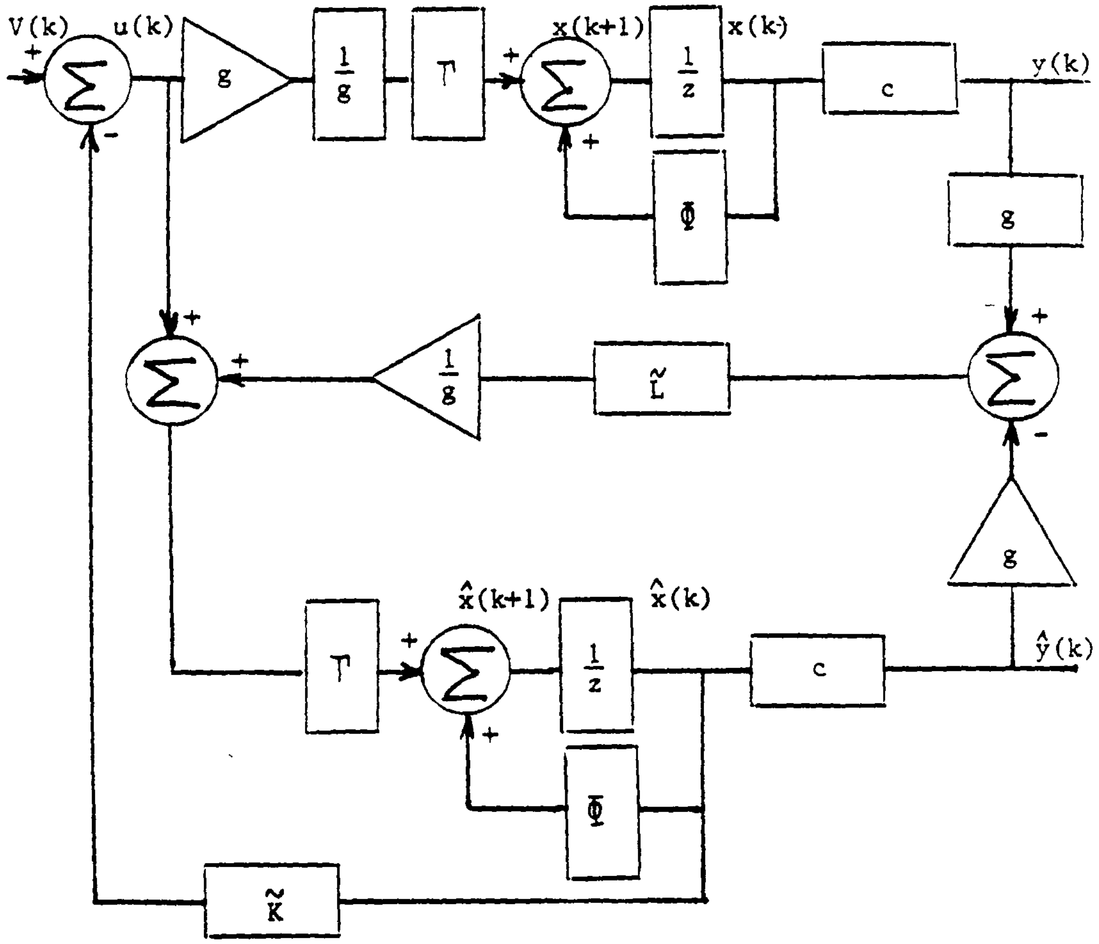
Comparing (2) with (4) and (3) with (5), the systems do not have the same eigenvalues due to this scaling and must be scaled to produce equivalency.



g = integer 2048/10 volts dc

Figure 12 Experimental Setup with Unavoidable Scaling

Figure 13 is developed from Figure 12 by adding the proper scaling factors (denoted for clarity by triangles) to change equations (4) and (5) to be consistent with equations (2) and (3).



denotes added scaling ▷

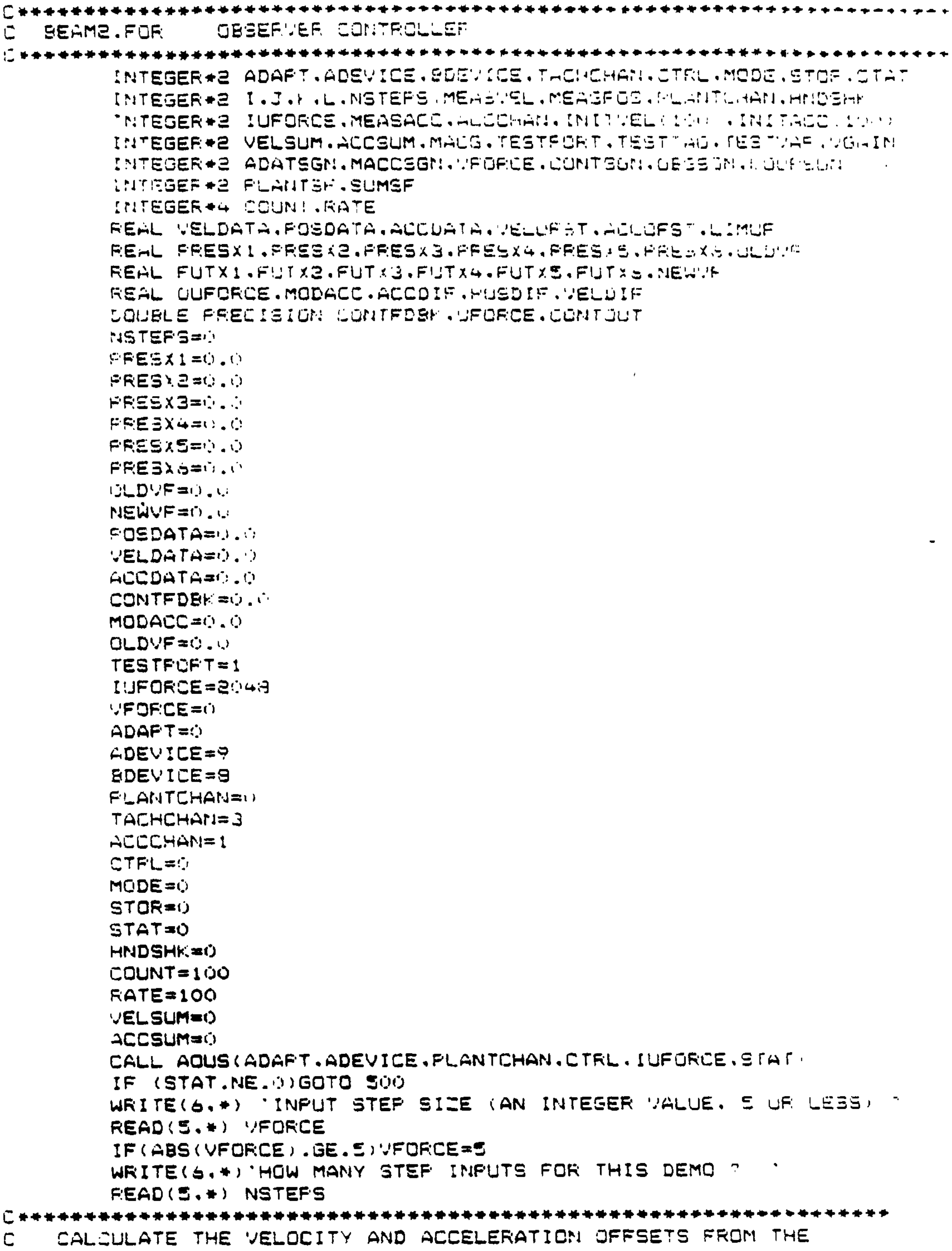
Figure 13 Rescaled Experimental System

Plant sensor offsets are cancelled in the experiement by repetitively sampling the A/Ds during program startup (no input to the plant), averaging the result and subtracting this from the sensor output each time it is read during program execution.

The Fortran Program to implement the Observer-Controller of this chapter is found in Figure 14 (3 pages). The functional order of such a program is critical to accomplish what is desired of the program. A functional synopsis of Figure 14 is listed below.

1. Initialize variables, measure plant offsets and enter main loop.
2. Generate new input; read plant sensors as close together in time as possible (i.e., read A/Ds and binary input).
3. Compute and generate new output from new input and present states (i.e., update D/A).
4. Compute all future states using new inputs and present states.
5. Assign future states to present states.
6. Return to 2.

This program was not written to be run synchronously with an update interrupt because it is the only code running in the PC at any one time. Relatively constant computation times produce a 10 msec time line with about ±5% breathing. Calls to the data acquisition adapter occupy 6 msec of the 10 msec total time line while the iterative calculations require the remaining time. The IBM PC data acquisition adapter macros which produce the I/O operations are somewhat inefficient for our purposes because at each READ/WRITE call, the PC register information must be pushed onto the 8086 stack to service the acquisition adapter, then popped when the operation is complete, only to be repushed microseconds later at a subsequent call for the next sensor (tachometer, accelerometer then binary encoder). Efforts will be made in the future to write code in assembler and to avoid these unnecessary delays.

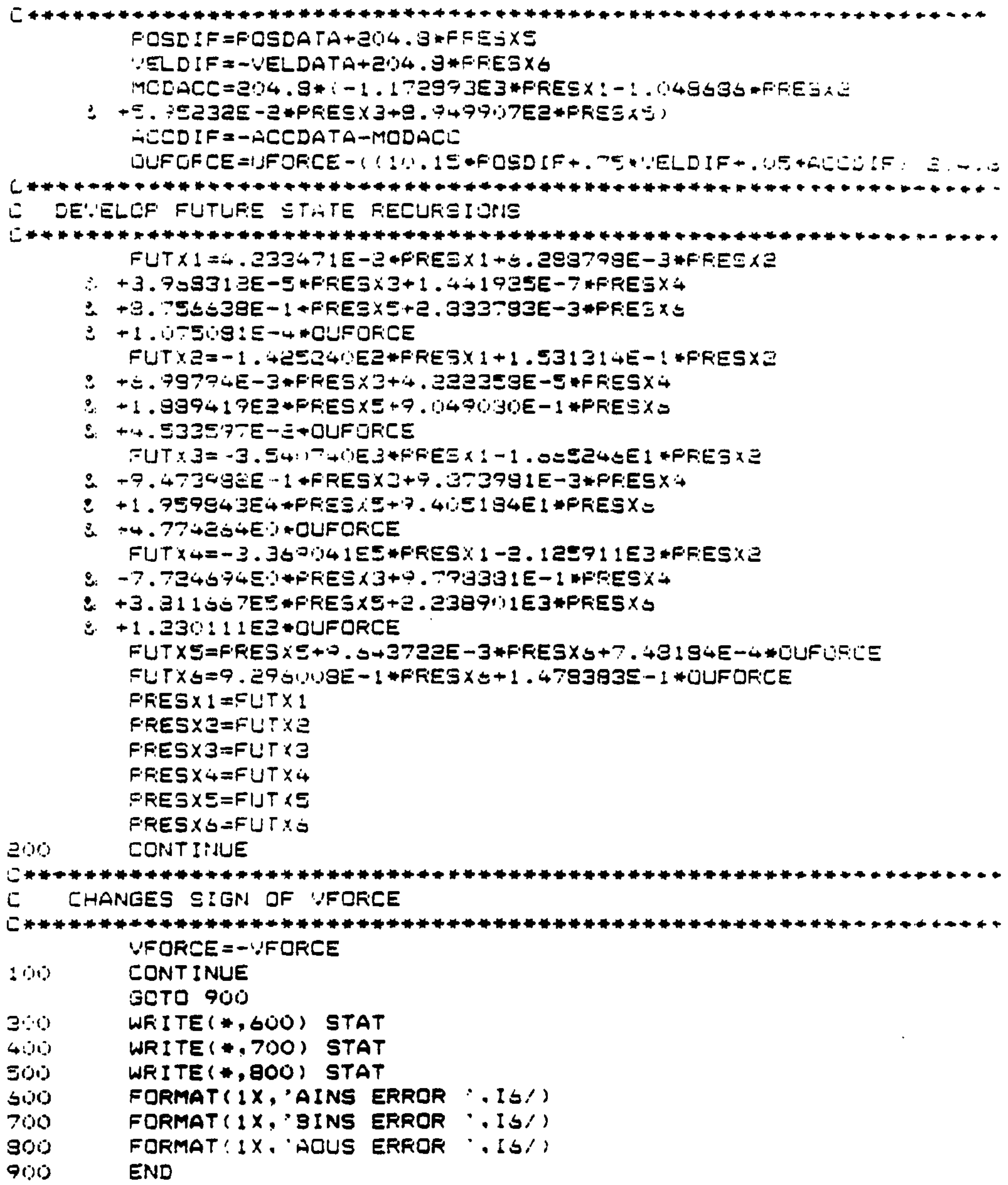


```
C*****************************************************************
C  BEAM2.FOR     OBSERVER CONTROLLER
C*****************************************************************
        INTEGER*2 ADAPT.ADEVICE.BDEVICE.TACHCHAN.CTRL.MODE.STOR.STAT
        INTEGER*2 I.J.K.L.NSTEPS.MEASVEL.MEASPOS.PLANTCHAN.HNDSHK
        INTEGER*2 IUFORCE.MEASACC.ACCCHAN.INITVEL(100).INITACC(100)
        INTEGER*2 VELSUM.ACCSUM.MACG.TESTPORT.TESTTAG.TESTVAF.VGAIN
        INTEGER*2 ADATSGN.MACCSGN.VFORCE.CONTSGN.OBSSGN.OUFSGN
        INTEGER*2 PLANTSF.SUMSF
        INTEGER*4 COUNT.RATE
        REAL VELDATA.POSDATA.ACCDATA.VELOFST.ACCOFST.LIMUF
        REAL PRESX1.PRESX2.PRESX3.PRESX4.PRESX5.PRESX6.OLDVF
        REAL FUTX1.FUTX2.FUTX3.FUTX4.FUTX5.FUTX6.NEWVF
        REAL OUFORCE.MODACC.ACCDIF.POSDIF.VELDIF
        DOUBLE PRECISION CONTFDBK.UFORCE.CONTOUT
        NSTEPS=0
        PRESX1=0.0
        PRESX2=0.0
        PRESX3=0.0
        PRESX4=0.0
        PRESX5=0.0
        PRESX6=0.0
        OLDVF=0.0
        NEWVF=0.0
        POSDATA=0.0
        VELDATA=0.0
        ACCDATA=0.0
        CONTFDBK=0.0
        MODACC=0.0
        OLDVF=0.0
        TESTPORT=1
        IUFORCE=2048
        VFORCE=0
        ADAPT=0
        ADEVICE=9
        BDEVICE=8
        PLANTCHAN=0
        TACHCHAN=3
        ACCCHAN=1
        CTRL=0
        MODE=0
        STOR=0
        STAT=0
        HNDSHK=0
        COUNT=100
        RATE=100
        VELSUM=0
        ACCSUM=0
        CALL AOUS(ADAPT.ADEVICE.PLANTCHAN.CTRL.IUFORCE.STAT)
        IF (STAT.NE.0)GOTO 500
        WRITE(6.*) 'INPUT STEP SIZE (AN INTEGER VALUE. 5 OR LESS) '
        READ(5.*) VFORCE
        IF(ABS(VFORCE).GE.5)VFORCE=5
        WRITE(6.*)'HOW MANY STEP INPUTS FOR THIS DEMO ?  '
        READ(5.*) NSTEPS
C***************************************************************
C   CALCULATE THE VELOCITY AND ACCELERATION OFFSETS FROM THE
```

Figure 14   Observer-Controller Fortran Program

```
C       PLANT SENSORS AND THEN SUBTRACT THE OFFSET FROM THE
C         RETURNS DURING THE REAL TIME ROUTINE CALLS
C*****************************************************************
         CALL AINM(ADAPT.ADEVICE.TACHCHAN.CTRL.MODE.STOP.COUNT.RATE
     & INITVEL(1).STAT)
         IF(STAT.NE.0)GOTO 300
         DO 10 K=1.100.1
         VELSUM=VELSUM+INITVEL(K)-2048
10       CONTINUE
         VELOFST=VELSUM/100
         CALL AINM(ADAPT.ADEVICE.ACCCHAN.CTRL.MODE.STOP.COUNT.RATE.
     & INITACC(1).STAT)
         IF(STAT.NE.0)GOTO 300
         DO 20 L=1.100.1
         ACCSUM=ACCSUM+INITACC(L)-2048
20       CONTINUE
         ACCOFST=ACCSUM/100
         DO 100 I=1.NSTEPS
         DO 200 J=1.500
C****************************************************************
C         25 HZ INPUT FILTER
C****************************************************************
         NEWVF=.061*VFORCE+.39*OLDVF
C****************************************************************
C   SEPARATE CALLS TO THE DATA ACQUISITION ADAPTER FOR PLANT
C        HUB RATE. HUB POSITION AND BEAM TIP ACCELERATION
C****************************************************************
         CALL AINS(ADAPT.ADEVICE.TACHCHAN.CTRL.MEASVEL.STAT)
         IF (STAT.NE.0) GOTO 300
         CALL AINS(ADAPT.ADEVICE.ACCCHAN.CTRL.MEASACC.STAT)
         IF (STAT.NE.0) GOTO 300
         CALL SINS(ADAPT.SDEVICE.HNDSHK.MEASPOS.STAT)
         IF (STAT.NE.0) GOTO 400
         VELDATA=(MEASVEL-2048-VELOFST)*2.45
         ACCDATA=MEASACC-2048-ACCOFST
         POSDATA=(MEASPOS/16)*.314
C****************************************************************
C    DEVELOP CONTROLLER STATE FEEDBACK AND SUBTRACT THIS FROM
C     THE INPUT. LIMIT ONLY IUFORCE WHICH IS THE PLANT ERROR.
C****************************************************************
         CONTFDBK=(1.833829E1*PRESX1-1.097366E-1*PRESX2
     & -2.299116E-3*PRESX3+1.488956E-5*PRESX4
     & +1.93973E1*PRESX5+7.421781E-1*PRESX6)
         UFORCE=NEWVF-CONTFDBK
         LIMUF=-UFORCE*204.8
         IF(LIMUF.GE.2000)LIMUF=2000
         IF(LIMUF.LE.(-2000))LIMUF=(-2000)
         IUFORCE=LIMUF+2048
         CALL AOUS(ADAPT.ADEVICE.PLANTCHAN.CTRL.IUFORCE.STAT)
         IF(STAT.NE.0)GOTO 500
         OLDVF=NEWVF
C****************************************************************
C      TAKE DIFFERENCES BETWEEN PLANT HUB RATE. HUB POSITION.
C  AND BEAM TIP ACCELERATION AND RESPECTIVE MODELED PARAMETERS
C              FROM THE OBSERVER AND SCALE BY L-tilde
```

Figure 14   Observer-Controller Fortran Program (Continued)



```
C*****************************************************************
         POSDIF=POSDATA+204.8*FRESX5
         VELDIF=-VELDATA+204.8*PRESX6
         MODACC=204.8*(-1.172893E3*PRESX1-1.048686*PRESX2
     & +5.75232E-2*PRESX3+8.949907E2*PRESX5)
         ACCDIF=-ACCDATA-MODACC
         OUFORCE=UFORCE-((10.15*POSDIF+.75*VELDIF+.05*ACCDIF)) 2.4.6
C*****************************************************************
C   DEVELOP FUTURE STATE RECURSIONS
C*****************************************************************
         FUTX1=4.233471E-2*PRESX1+6.288798E-3*PRESX2
     & +3.968318E-5*PRESX3+1.441925E-7*PRESX4
     & +8.756638E-1*PRESX5+2.333783E-3*PRESX6
     & +1.075081E-4*OUFORCE
         FUTX2=-1.425240E2*PRESX1+1.531314E-1*PRESX2
     & +6.99794E-3*PRESX3+4.222358E-5*PRESX4
     & +1.889419E2*PRESX5+9.049030E-1*PRESX6
     & +4.533597E-3*OUFORCE
         FUTX3=-3.540740E3*PRESX1-1.665246E1*PRESX2
     & +9.473988E-1*PRESX3+9.373981E-3*PRESX4
     & +1.959843E4*PRESX5+9.405184E1*PRESX6
     & +4.774264E0*OUFORCE
         FUTX4=-3.369041E5*PRESX1-2.125911E3*PRESX2
     & -7.724694E0*PRESX3+9.778331E-1*PRESX4
     & +3.311667E5*PRESX5+2.238901E3*PRESX6
     & +1.230111E2*OUFORCE
         FUTX5=PRESX5+9.643722E-3*PRESX6+7.48194E-4*OUFORCE
         FUTX6=9.296008E-1*PRESX6+1.478383E-1*OUFORCE
         PRESX1=FUTX1
         PRESX2=FUTX2
         PRESX3=FUTX3
         PRESX4=FUTX4
         PRESX5=FUTX5
         PRESX6=FUTX6
200      CONTINUE
C*****************************************************************
C   CHANGES SIGN OF VFORCE
C*****************************************************************
         VFORCE=-VFORCE
100      CONTINUE
         GOTO 900
300      WRITE(*,600) STAT
400      WRITE(*,700) STAT
500      WRITE(*,800) STAT
600      FORMAT(1X,'AINS ERROR ',I6/)
700      FORMAT(1X,'BINS ERROR ',I6/)
800      FORMAT(1X,'AOUS ERROR ',I6/)
900      END
```

Figure 14 Observer-Controller Fortran Program (Continued)

VII. EXPERIMENTAL RESULTS AND CONCLUSIONS

Program Beam2 in chapter VI is a simplification of Beam1 which has the capability of calling program variables to appear at the instrumentation port on the electronics chassis. Program Beam1 was used to access plant and observer variables to plot the results shown in this chapter. Plotting was performed by connecting the Y-axis of the X-Y plotter to the instrumentation port and connecting the X-axis to the plotter's internal time base.

Figure 1 compares the sampled plant encoder (shaft position) output with the observer state x_5 for a step input of 0.3 radian. Figure 2 compares the sampled plant tachometer (shaft rate) output with the observer state x_6 under the same input. The close correlation in each figure is expected because each of these estimated states is directly corrected by plant feedback.

Figure 3 is a comparison of the sampled beam tip accelerometer output and tip acceleration as developed in the observer. Although the low frequency correlation is good, there is significant departure at higher frequencies. Particularly curious is the 6.5hz oscillation which is most noticable between 1.0 and 1.5 seconds. The source of this mode is not clear at this time but it is expected to be due to unmodeled or phase-shifted modeled frequencies fed into the observer through the plant tip accelerometer. These appear as external

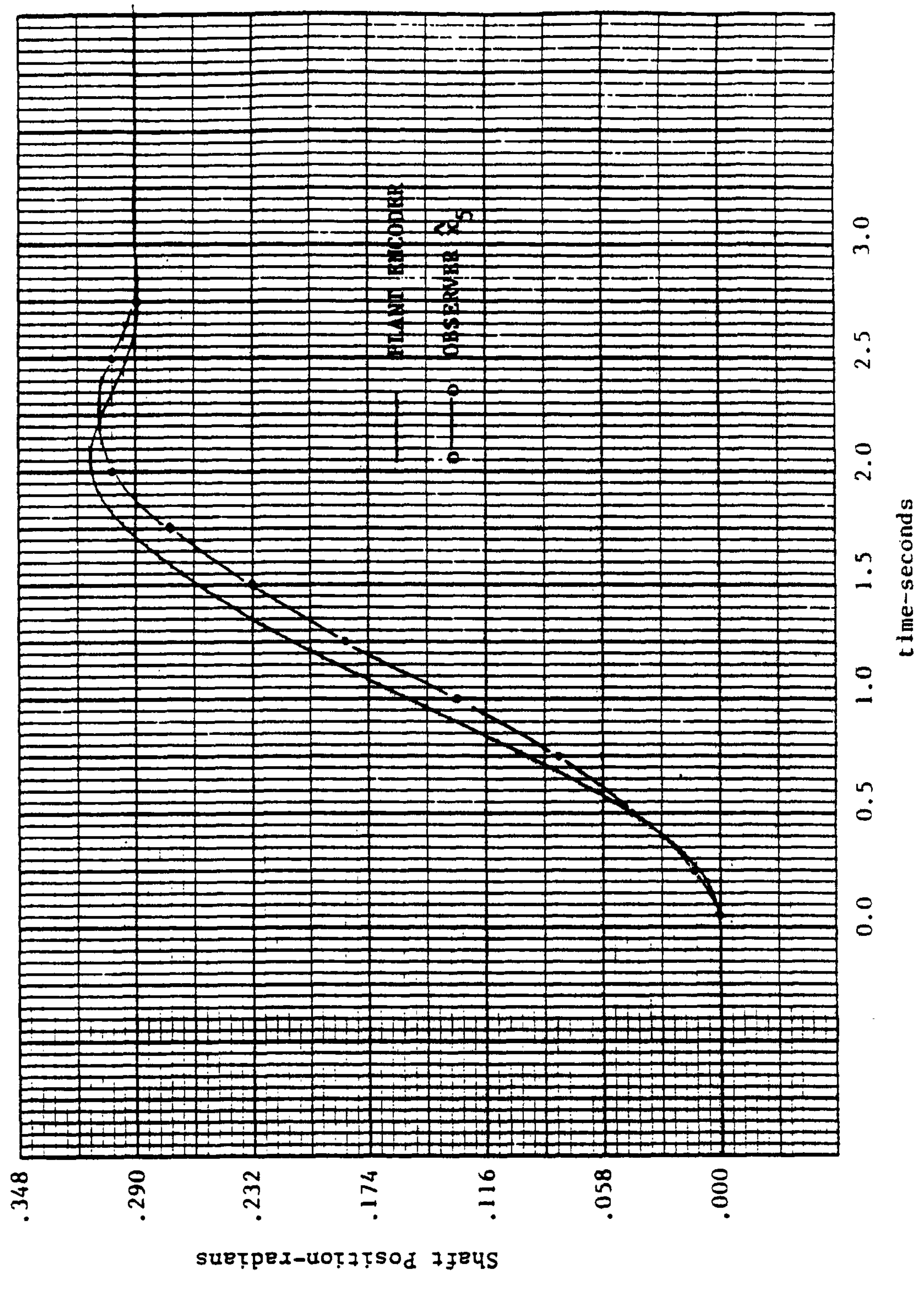


Figure 1 - Shaft Position Comparison

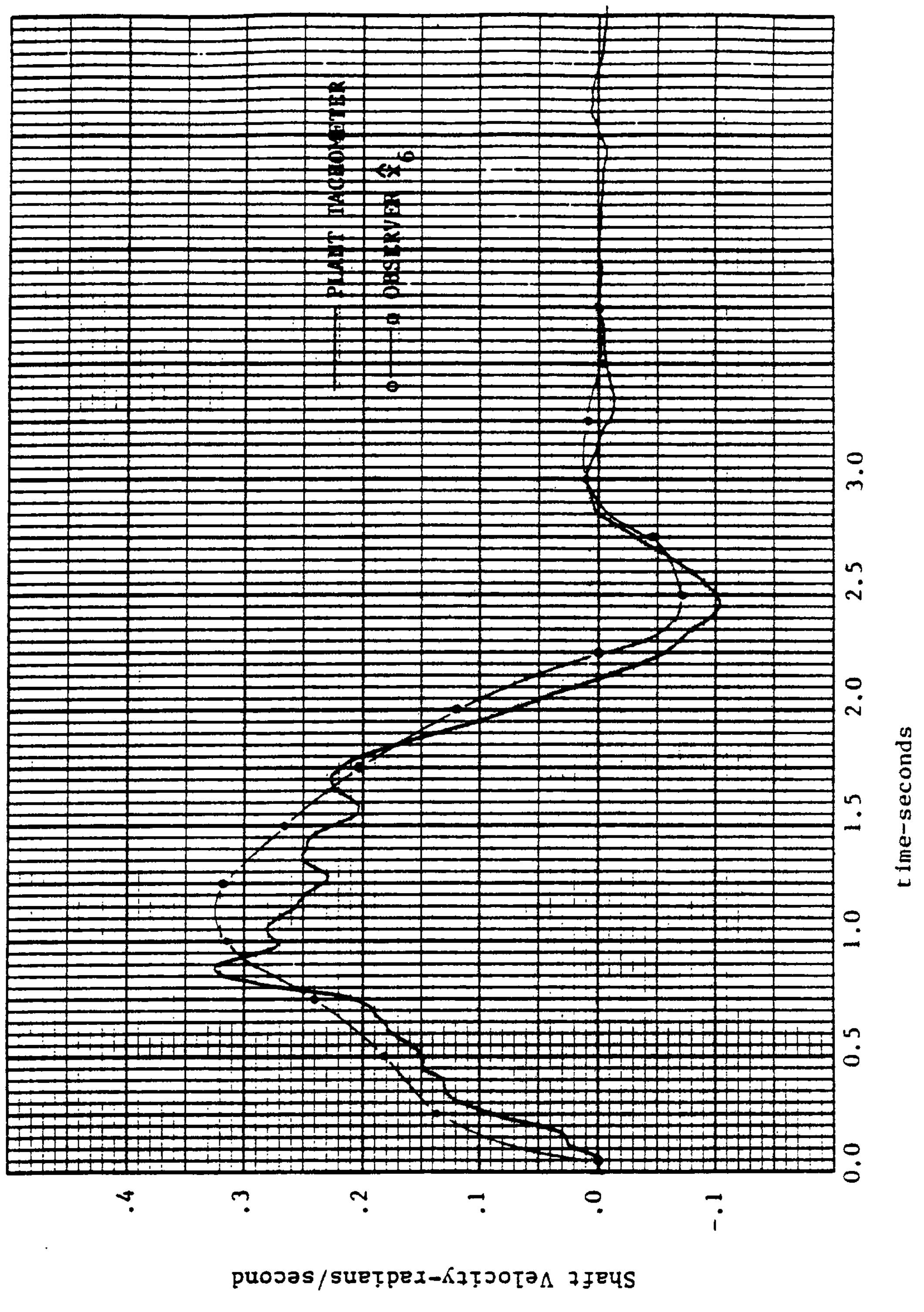


Figure 2 - Shaft Rate Comparison

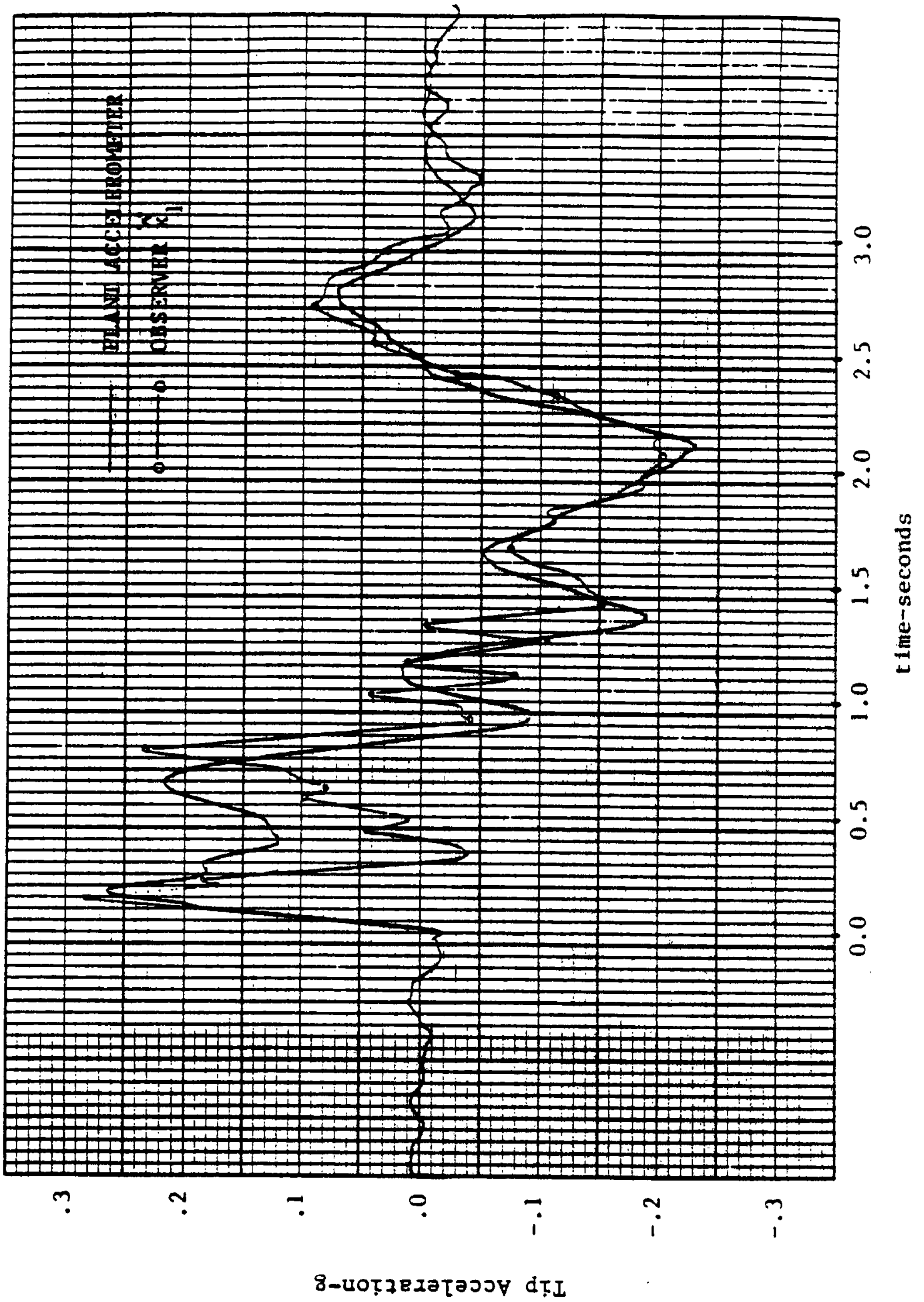


Figure 3 - Tip Acceleration Comparison

disturbances to the plant and see a radically different transfer function than does the normal input. Additional simulation has been planned to investigate this hypothesis. It is particularly interesting to note that the root locus of the transfer function from plant input to tip acceleration includes a complex pole pair (20.6hz) very close to the unit circle with angles of departure of 90° and 270°. This system goes unstable for a feedback gain greater than .03 (the experimental value is .01). The combination of this low gain and the large sampling phase shift at 20hz causes poor correction of the estimated beam states for high frequency inputs. These problems will be addressed in the future by increasing the sampling rate and by improving the observer output feedback performance.

Figure 4 compares the simulated step response with the measured data. Both of the curves are the observer x_1 state, the solid curve has no plant state correction (it is the lower half of Figure 2 on page 50). The circle curve is the closed loop observer-controller of Figure 3 on page 52. In Figure 4 of this chapter, the first radical departure from the system simulation is observed. The experimental system tip position lags the simulated system by about one second as the tip moves 0.3 radian. The tip overshoot and settling characteristics are identical for the simulated and experimental systems.

The lag is due to a combination of two factors: low control cost and high system friction. The DELIGHT.MaryLin optimization for the

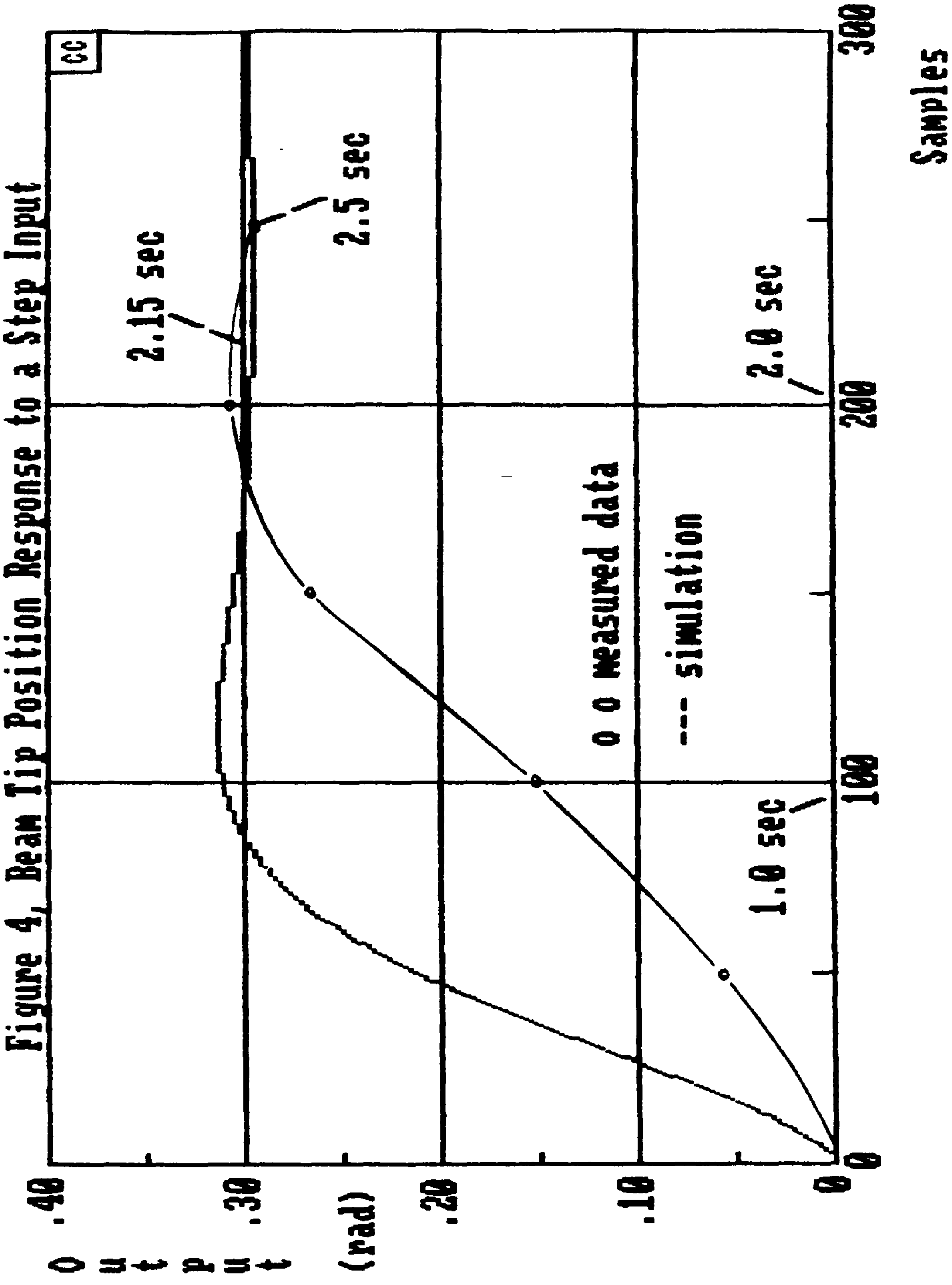


Figure 4 - Tip Position Comparison

given engineering specifications produced a system with very low torque requirements, low enough to be roughly what is needed to overcome friction. The friction acts as a non-linear plant disturbance which has not been taken into account in the model.

A solution to this problem has been developed. The plant tachometer can be used to close an analog control loop around the motor yielding a velocity source with a linear scale factor of r (rad/sec)/volt at the motor shaft. A new DELIGHT.MaryLin optimization will be needed to implement this design.

In summary, good control of a system utilizing an observer-controller can be achieved by observing normal design practice and the following cautions:

a. Sampling speed must be adequate to uniquely reproduce each expected modal frequency.

b. Phase shift between sampled plant sensors and estimated observer states must be minimized.

c. Friction must be "handled" through modeling or through plant modification, expecially in direct drive systems.

d. Engineering specifications to DELIGHT.MaryLin, in this optimization environment, may not be as obvious as they seem, e.g., modal time constants are important.

e. External disturbances do not affect the observer-controller in the same way they might affect a classical system. More attention must be paid to this issue.

The initial experiment of Cannon and Schmitz was used as a basis for this experiment. The chief difference in concept relates to the use of an absolute position tip sensor in the Cannon and Schmitz work and a tip accelerometer in this experiment. The intent of both experiments is to control tip position of 36 inch long beam. In practice, it is difficult to measure the tip position of such a beam (for obvious reasons) and, while a tip accelerometer is not the friendliest of sensors, it is easy to implement. Either sensor can be used to correct the beam state estimates providing effective system control.

Figure 5 compares an experimental result of Cannon and Schmitz, the 5 degree step response, with this experiment (the results in Figures 1-4 are for a step of 17.5°). Recalling the friction problems with this experiment and noting that the engineering specifications allowed DELIGHT.MaryLin t. overshoot and recover slowly, the responses

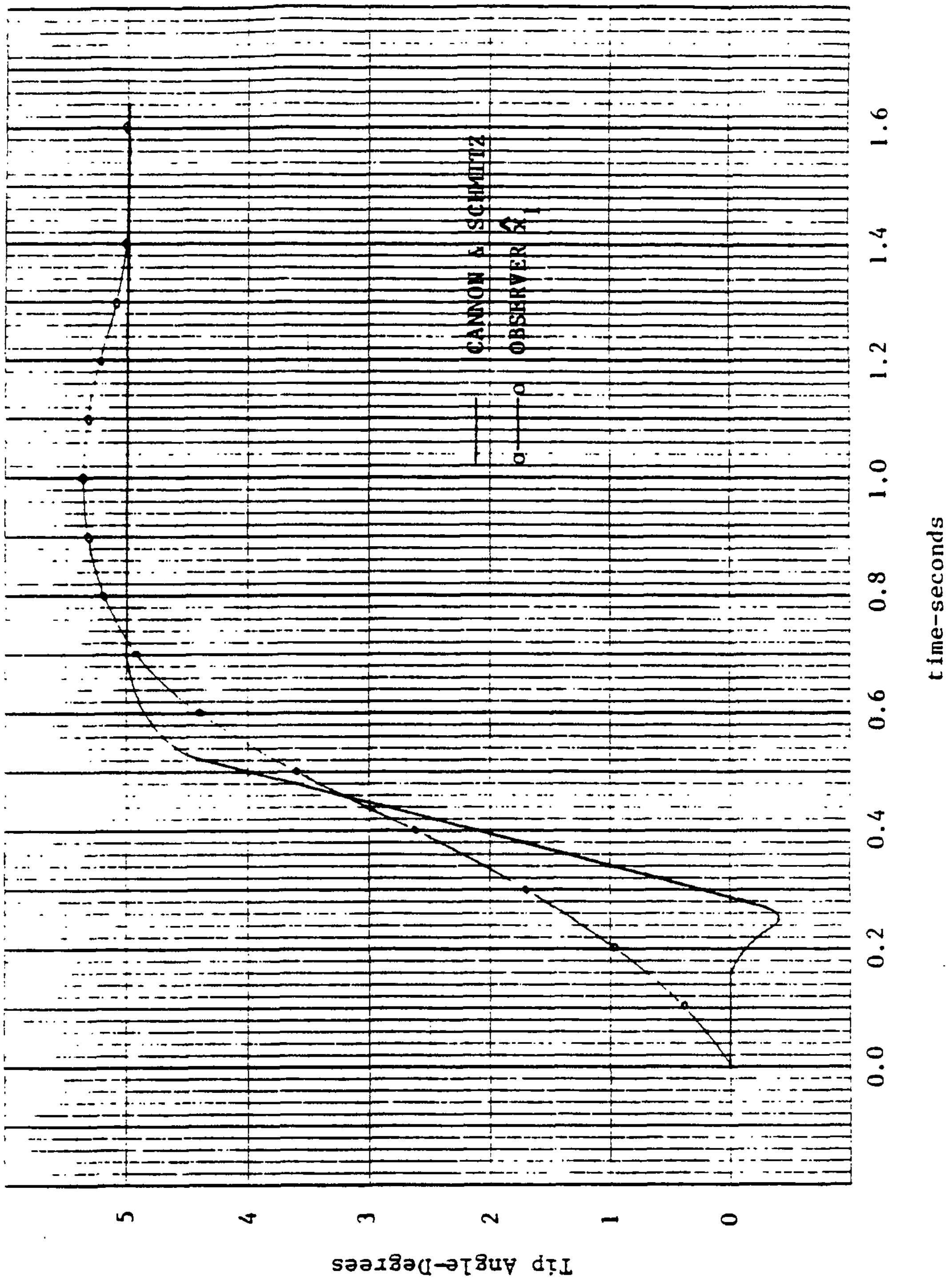


Figure 5 - Cannon and Schmitz Comparison

are not drastically different. The University of Maryland Arm response will be very similar to the arm of Cannon and Schmitz after the friction problem is fixed and a new DELIGHT.MaryLin optimization is complete.

The directions pursued by the SRC in the future include:

1. Modify the plant to make friction transparent.

2. Increase the sample rate of the PC.

3. Change the engineering specifications for a new DELIGHT.MaryLin optimization.

4. Improve the observer output feedback response.

5. Install piezo film on the arm to act as a bending sensor and also provide beam damping with a similar film.

6. Investigate distributed rate and acceleration sensors for the arm.

APPENDIX A

Derivation of First Resonant Mode Shift from Lumped Parameter Model

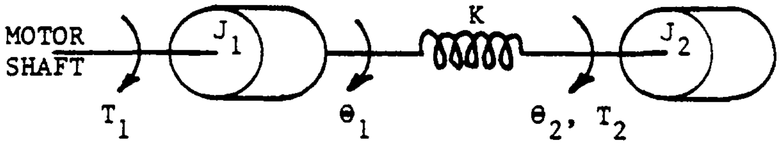


Figure A-1 Lumped Inertia Model of Plant

J_a = Calculated inertia of the motor shaft, motor and table.

J_b = Inertial portion of the flexible beam which is carried rigidly, i.e., does not decouple from the shaft, in the first bending mode.

J_1 = $J_a + J_b$, the inertia moving with the motor shaft.

J_2 = Inertial portion of the beam which is not carried rigidly, i.e., decouples from the shaft, in the first bending mode $(J_{beam} - J_b)$.

K = First bending mode beam stiffness.

T_1, T_2 = Respective input torques.

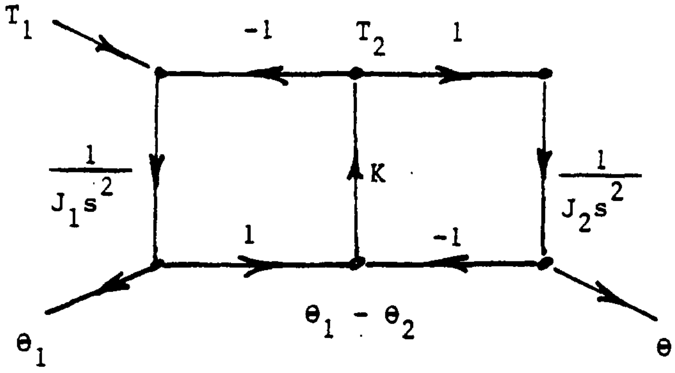


Figure A-2 Signal Flow Graph for Figure A-1

A.1 For fixed-free bending of the beam alone

$J_1 = \infty$

$T_1 = 0$

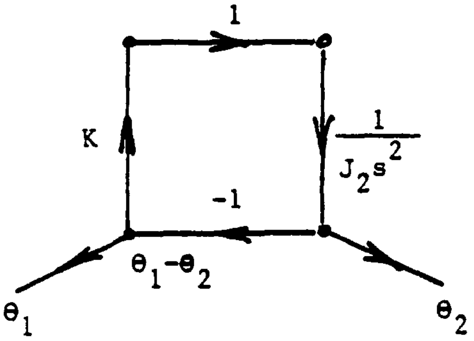


Figure A.1-1 Fixed-Free Signal Flow Graph

By Mason's rule:

$$\frac{\theta_2}{\theta_1} = \frac{K/J_2 s^2}{1 + K/J_2 s^2} = \frac{1}{(s^2 /(\sqrt{K/J_2})^2)+1}$$

As stated in the text, this mode was calculated and confirmed by measurement to be 2.7hz. In chapter IV, curve fitting of the voltage to hub position transfer function yields an adjusted motor corner of 7.3 rad/sec (J_a + J_b), while calculation using just the hub inertia (J_a) produces a motor corner of 21.0 rad/sec. The difference is the portion of beam inertia carried "rigidly" by the hub, or:

$$7.3 \text{ r/s } (J_a + J_b) = 21 \text{ r/s } (J_a)$$

$$J_b = (13.7/7.3)\ J_a = .188 \text{ lb-in-sec}^2\text{/rad}$$

then, J_2 is the total beam inertia (1.0 lb-in-sec^2/rad, calculated) less J_b or 0.81 lb-in-sec^2/rad.

A.2 Plant first resonant mode

$$J_1 = .1 + .188 \quad .29 \text{ lb-in-sec}^2\text{/rad}$$

$$J_2 = .81 \text{ lb-in-sec}^2\text{/rad}$$

By Mason's rule from figure 2:

$$\frac{\theta_2}{T_1} = \frac{K/J_1J_2s^4}{1 + K/J_1s^2 + K/J_2s^2}$$

$$= \frac{1/s^2(J_1 + J_2)}{(s^2/(\sqrt{K(J_1 + J_2)/J_1J_2})^2) + 1}$$

For K unchanged, the first resonant mode of the plant is approximately ($\sqrt{K/J_2}$)(J_2)/(J_1J_2/(J_1+J_2)) or 2.7 (.81/.212)= 10.3hz. The modal identification experiment shows this to be 9.0hz.

APPENDIX B

Development of Beam Tip Acceleration from Beam States

Referring to Figure 4 on page 24, states x_1 through x_4 may be expressed in Kailath's observer cannonical representation to produce

$$A = \begin{bmatrix} -a_1 & 1 & 0 & 0 \\ -a_2 & 0 & 1 & 0 \\ -a_3 & 0 & 0 & 1 \\ -a_4 & 0 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix},$$

$$c = \quad [1\ 0\ 0\ 0]$$

$$y = cx = [1\ 0\ 0\ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Note that for this system $u = x_5$. Then,

$\dot{y} = c\dot{x}$, where $\dot{x} = Ax + bu$

and,

$\ddot{y} = c\ddot{x}$, where $\ddot{x} = A\dot{x} + bu = A(Ax + bu) + bu$.

Only the first row of the result is needed

$$\ddot{x} = \begin{bmatrix} a_1^2-a_2 & -a_1 & 1 & 0 \\ & & & \\ & & & \end{bmatrix} x + \begin{bmatrix} b_2 \\ \\ \end{bmatrix} u + \begin{bmatrix} 0 \\ \\ \end{bmatrix} u$$

$$\ddot{y} = c\ddot{x} = [a_1^2-a_2 \quad -a_1 \quad 1 \quad 0 \quad b_2 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

APPENDIX C

```
C        ENCANG.FOR
C*********************************************************************
         SUBROUTINE ENCANG(RAWVAL)
         INTEGER*2 ADAPTER.DEVICE,HNDSHK.RAWVAL.STAT
         ADAPT=0
         DEVICE=8
         HNDSHK=0
         STAT=0
         CALL BINS (ADAPT.DEVICE,HNDSHK.RAWVAL.STAT)
         IF (RAWVAL.GE.2048) RAWVAL=2048
         IF (RAWVAL.LE.(-2048)) RAWVAL=(-2048)
         RAWVAL=RAWVAL*(-1)/2
         GOTO 500
500      RETURN
         END
```

```
C        IDENTST.FOR
C        SOFT CLOSED LOOP WITH BINS AND AOUS
C***********************************************************************
         INTEGER*2 RAWDTA.STATO.RAWDAT,FDBK
         INTEGER*2 ADAPT.DEVICE,CHANLOO.CHANLO1.CTRL.STOR.STAT1.FLAG
         ADAPT=0
         DEVICE=9
         CHANLOO=0
         CHANLO1=1
         CTRL=0
         STATO=0
         STAT1=0
         FLAG=0
100      CALL ENCANG(RAWDTA)
         RAWDAT=(RAWDTA+2048)
         FDBK=((RAWDTA/10)+2048)
         CALL AOUS(ADAPT.DEVICE,CHANLO1.CTRL,RAWDAT.STAT1)
         CALL AOUS(ADAPT,DEVICE,CHANLOO,CTRL,FDBK.STATO)
         FLAG=FLAG + 1
         IF (FLAG.EQ.1) WRITE (*,200) RAWDTA
200      FORMAT (1X,I6)
         IF (FLAG.EQ.100) FLAG=0
         GOTO 100
800      END
```

Low Bandwidth Closed Loop for Modal Identification

BIBLIOGRAPHY

(1) Karl J. Astrom, Bjorn Wittenmark, Computer Controlled Systems - Theory and Design, 1st ed. (Englewood Cliffs, N.J. : Prentice-Hall, Inc., 1984), pp 189-192

(2) Robert H. Cannon, Jr., Eric Schmitz, Initial Experiments on the End-Point Control of a Flexible One-Link Robot, International Journal of Robotics Research, Vol. 3, No. 3, Fall, 1984, pp 62-75

(3) Thomas Kailath, Linear Systems, 1st ed. (Englewood Cliffs, N.J. : Prentice-Hall, Inc., 1980), pp 31-313

(4) W. T. Nye and A. L. Tits, An Application - Oriented, Optimization - Based Methodology for Interactive Design of Engineering Systems, International Journal of Control, Vol. 43, No. 6, 1986, pp 1693-1721

(5) Bill Nye, Michael Ko-Hui Fan and Andre' Tits, DELIGHT.MaryLin User's Guide (Unfinished), 1985