N90-10088

FLEXIBLE MISSILE AUTOPILOT DESIGN STUDIES WITH PC-MATLAB/386

By

Michael J. Ruth
Johns Hopkins University/Applied Physics Laboratory
Laurel, Maryland

ABSTRACT

Development of a responsive, high-bandwidth missile autopilot for airframes which have structural modes of unusually low frequency presents a challenging design task. Such systems are viable candidates for modern, state-space control design methods. The PC-MATLAB interactive software package provides an environment well-suited to the developement of candidate linear control laws for flexible missile autopilots. The strengths of MATLAB include: (1) Exceptionally high speed -- MATLAB's version for 80386-based PC's offers benchmarks approaching minicomputer and mainframe performance; (2) Ability to handle large design models of several hundred degrees of freedom, if necessary; and (3) Broad extensibility through user-defined functions. To characterize MATLAB capabilities, a simplified design example is presented. This involves interactive defintion of an observer-based state-space compensator for a flexible missile autopilot design task. MATLAB capabilities and limitations, in the context of this design task, are then summarized.

PRECEDING PAGE BLANK NOT FILMED

PAGE 138 INTENTIONALLY BLANK

FLEXIBLE MISSILE AUTOPILOT DESIGN STUDIES WITH PC-MATLAB/386

Michael J. Ruth
Johns Hopkins University / Applied Physics Laboratory
Laurel, Maryland

Workshop on Computational Aspects in the Control of Flexible Systems

12-14 July 1988
Williamsburg, Virginia

PRESENTATION OVERVIEW

1. Introduction
2. MATLAB Background
3. Characteristics of MATLAB Environment
4. Classical Control Capabilities
5. Modern Control Design Example
6. Summary

INTRODUCTION

- JHU/APL acts as technical direction agent for US Navy weapon system programs

- A key task of APL's Guidance, Control, and Navigation Systems Group is the evaluation or conceptual design of missile guidance and control systems

- Analysis and design work requires a <u>flexible</u>, <u>interactive</u> linear modeling tool

- PC-MATLAB resident on 80386 engineering work-stations provides such a tool

- Work presented here shows general attributes of MATLAB, demonstrating use of PC-MATLAB/386 for linear design of a flexible missile autopilot

MATLAB BACKGROUND

- MATLAB (MATrix LABoratory) provides an interactive, matrix-oriented environment
- MATLAB is based on the EISPACK and LINPACK routines for matrix computations
- PC-MATLAB/386 is a high-performance MATLAB implementation for 80386-based workstations
- MATLAB built-in functions, plus higher-level functions developed for control system calculations, allow for effective controls design studies

HARDWARE AND SOFTWARE CONFIGURATION

- COMPAQ 386/20 computer
- Weitek 1167 numeric coprocessor
- PC-MATLAB/386 with Control Systems Toolbox

PC-MATLAB/386 ATTRIBUTES

- Interactive, high-level command environment

- Very high processing speed

- Easy extensibility via user-defined functions

A MATLAB INTERACTIVE COMMAND LINE EXAMPLE

```
>> k = lqr(a,b,q,rho*r); eig(a-b*k), y = step(a-b*k,b,c,d,1,t); plot(t,y);
```

- The <u>single line</u> above, typed at the MATLAB command line prompt, does several things:
  - Computes a quadratic regulator gain vector
  - Displays the closed-loop eigenvalues -- often useful for confirming that actuator bandwidth requirements are not excessive
  - Computes and plots a unit step response
- By varying the control cost (rho) above, a very large family of compensators may quickly be considered
- The above command line suggests the power and utility available from a high-level, interactive matrix language

PC-MATLAB/386 PROCESSING SPEED

- MATLAB's LINPACK Benchmark: 460 double precision KFLOPS

- This processing speed is:
  - 25 x faster than standard PC/AT
  - 6 x faster than Mac II
  - 3 x faster than MicroVax II

- Implication: the fast response time resulting from such performance allows for truly interactive design iterations on complex control laws

MATLAB EXTENSIBILITY

- User-defined functions may be developed through creation of simple text files

- Some typical user-defined functions:
  - Frequency-response plotting routines
  - Application-specific linear transformations
  - Multivariable Nyquist criterion

- Complex state-space or transfer-function models also defined through user text files

AN EXAMPLE OF A USER-DEFINED COMMAND FILE

- **Below command set calculates and plots the maximum and minimum singular values of a plant and observer-based compensator, for a loop broken at plant input**

```
function [smin,smax] = svdinput(a,b,c,kcon,kobs,w);
%
jay = sqrt(-1);
[nn,xx]=size(a); i2=eye(nn); [ng,xx]=size(c*a*b); phi = '(s*i2-a)';
for i = 1:nc;
   s = w(i)*jay; phieval = eval(phi);
   gs = c/phieval*b; ks = kcon / (phieval+b*kcon+kobs*c) * kobs;
   xx=svd(ks*gs); smin(i)=xx(ng); smax(i)=xx(1);
end;
%
%
% convert to decibels and plot output
%
smin=20*log10(smin); smax=20*log10(smax);
semilogx(w,smin,w,smax,'r--'); grid;
title('Max and Min Singular Values; Loop Broken at Plant Input ');
xlabel('Frequency  (rad/sec)'); ylabel('Magnitude  (db)');
```

- **Procedure requires only eleven lines of executable MATLAB code**

CLASSICAL CONTROL CAPABILITIES

- Frequency response
- Root locus
- Nyquist plots
- Development of dynamic compensators (lead-lag, notch filters, etc)

MODERN CONTROL DESIGN EXAMPLE

- Design plant describes tactical missile at a high-altitude flight condition
- Design plant includes single-plane rigid-body dynamics and effect of first flexible mode on sensed pitch rate
- Objective is to develop an autopilot to track commanded accelerations
- Design challenge is to achieve high closed-loop bandwidth in presence of low-frequency bending modes

DESIGN APPROACH

- Establish design goals for closed-loop responsiveness and stability
- Develop full-state feedback (LQR) gains for design plant
- Define linear observer to reconstruct full state vector
  - Use "robust observer" design (Doyle and Stein, 1979 IEEE Transactions on Automatic Control)
  - Adjust observer gains to recover original LQR loop transfer in desired frequency range

DESIGN PLANT MODEL

- Fifth-order state vector $\underline{x}$; $\dot{x} = \underline{A}\underline{x} + \underline{b}u$

- $\underline{x} = [q_r \quad q_r/s \quad a/s \quad q_f/s \quad q_f]$

- First three state variables are associated with rigid-body airframe; the last two describe flexible mode dynamics

- Rate gyro measurement: $[1\ 0\ 0\ 0\ 1] * \underline{x}$

- (Integrated) accelerometer measurement: $[0\ 0\ 1\ 0\ 0] * \underline{x}$

$$\mathbf{A} = \begin{bmatrix} 0 & -2.3557e{+}02 & 1.7967e{+}02 & 0 & 0 \\ 1.0000e{+}00 & 0 & 0 & 0 & 0 \\ 0 & 2.6158e{+}00 & -1.9951e{+}00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000e{+}00 \\ 0 & 0 & 0 & -2.4649e{+}04 & -3.1400e{+}00 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -2.8031e{+}02 \\ 0 \\ 9.2587e{-}02 \\ 0 \\ 3.0723e{+}02 \end{bmatrix}$$

SOME OBSERVATIONS ON DESIGN PLANT MODEL

- Feedback of the first three states describes a very standard (rigid-body) autopilot topology, used by tactical missiles since 1950's

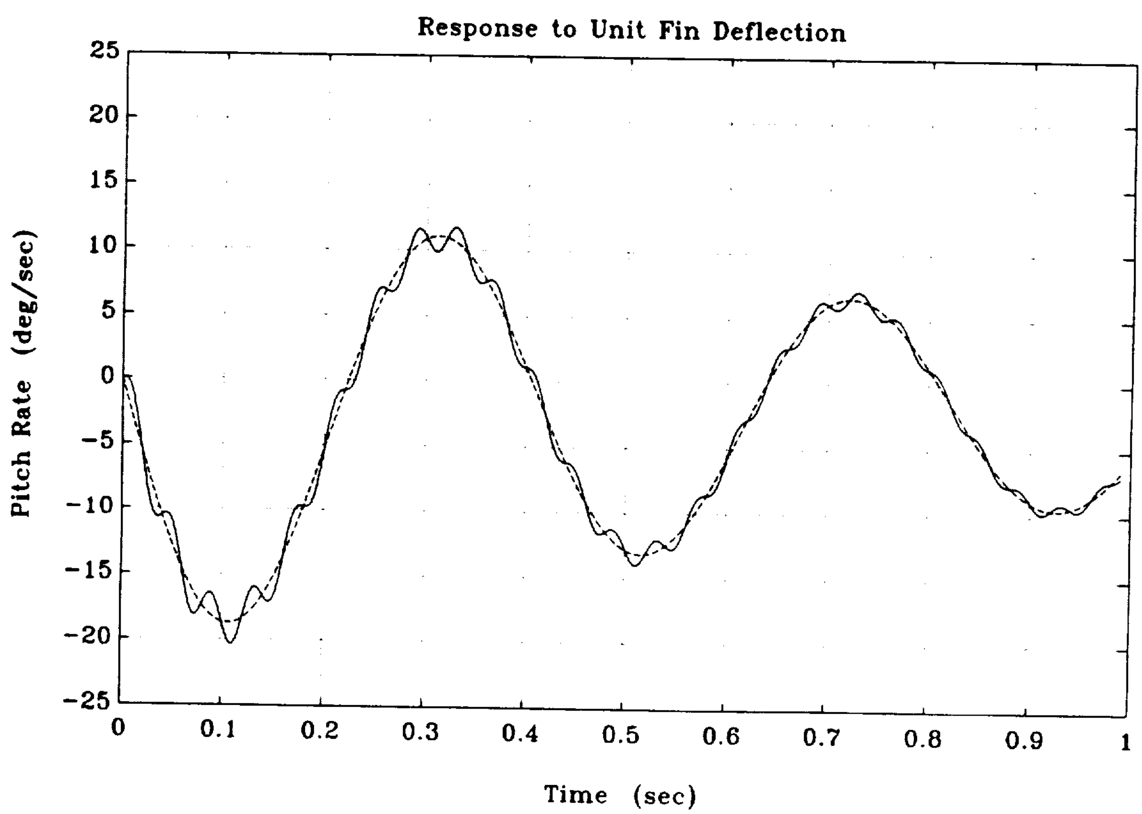
- Open-loop plant is characterized by lightly damped airframe (weathercock) poles, and by bending mode poles

  - Airframe pole frequency lies at nominal 2.5 Hz

  - Bending mode has nominal 25 Hz natural frequency

- Desired autopilot crossover frequency here will lie near the bending mode frequency

EFFECT OF STRUCTURAL MODE ON SENSED PITCH RATE (RATE GYRO MEASUREMENT)



CONTROLLABILITY AND OBSERVABILITY PROPERTIES OF PLANT

- System ($\underline{A}$,$\underline{b}$) is controllable

- System is unobservable if rate gyro alone, or accelerometer alone, is used as the measurement to reconstruct state vector

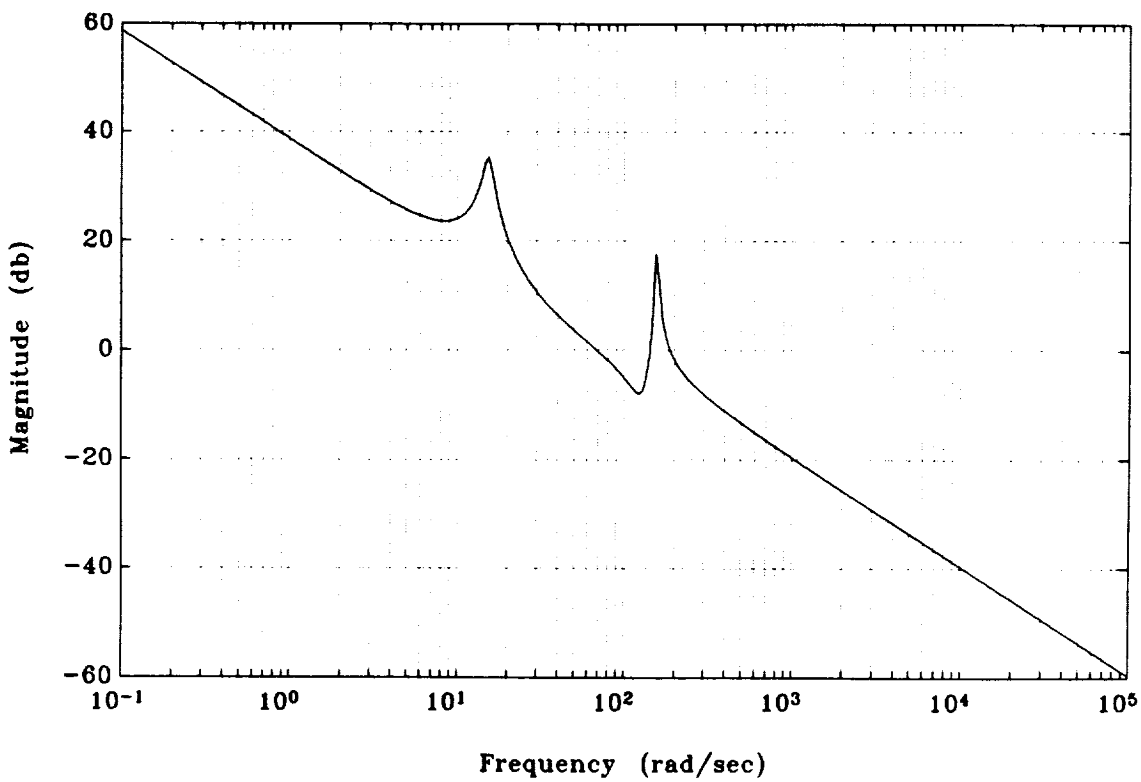
- Both sensor outputs thus should be used in the observer design

- Approach taken for this application:

  - Define a (non-square) design plant having one input (fin deflection) and two independent outputs (gyro and accelerometer)

  - Use extensions of loop transfer recovery (Williams and Madiwale, 1985 ACC) valid for non-square systems

FREQUENCY RESPONSE OF FULL-STATE FEEDBACK (LQR) SYSTEM (LOOP BROKEN AT PLANT INPUT)



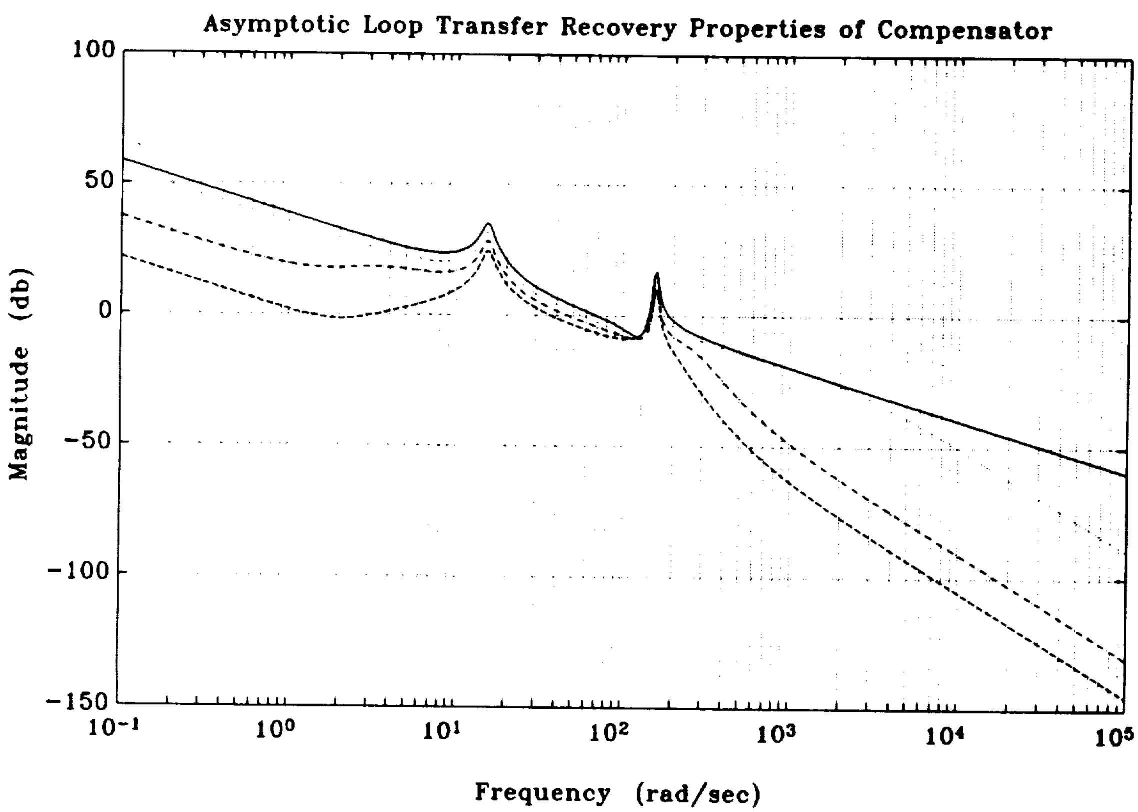
OBSERVATIONS ON LOOP TRANSFER RECOVERY PROCEDURE

- For this application, recovery at both the (rigid-body) airframe and bending mode frequencies may only be achieved with very high observer gains

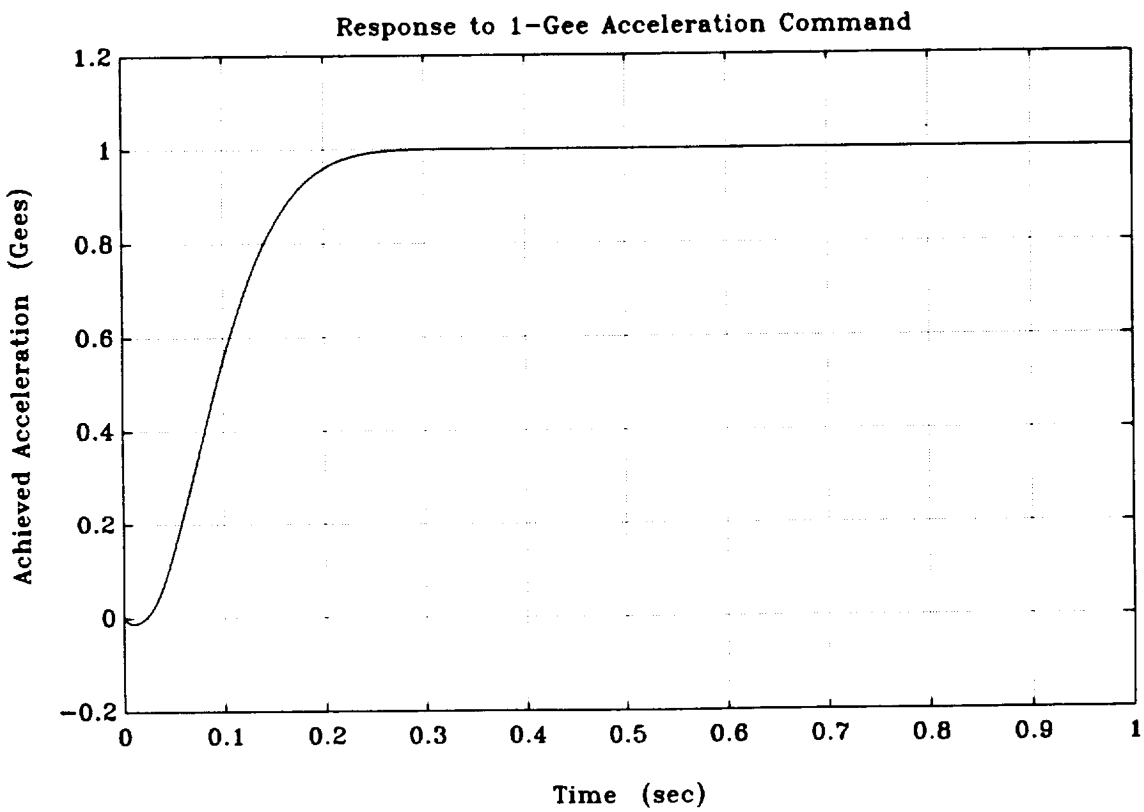
- For practical ranges of observer gains, recovery at airframe frequencies is obtained at the cost of lessened robustness in the structural mode frequency range

- Use of a set of user-defined MATLAB files, to implement a range of observer gain calculations, makes evaluation of this robustness tradeoff straightforward

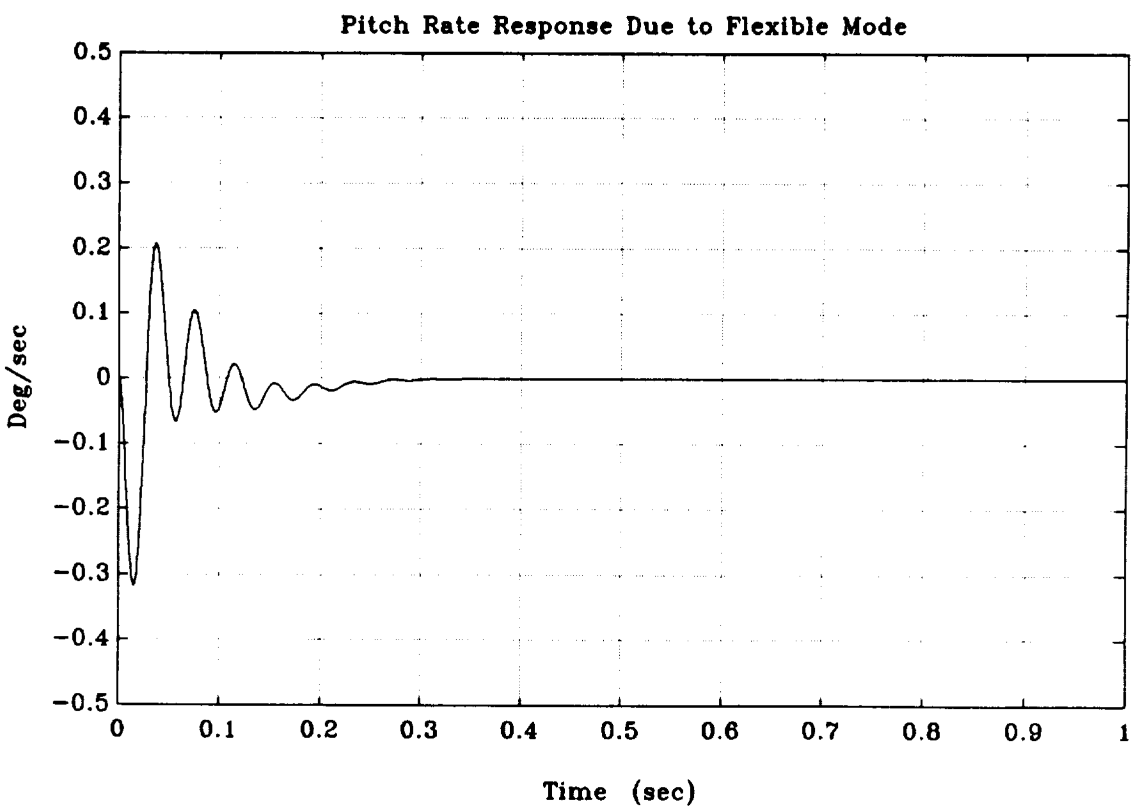
RECOVERY OF DESIRED FULL-STATE FEEDBACK SYSTEM WITH MODEL-BASED COMPENSATOR



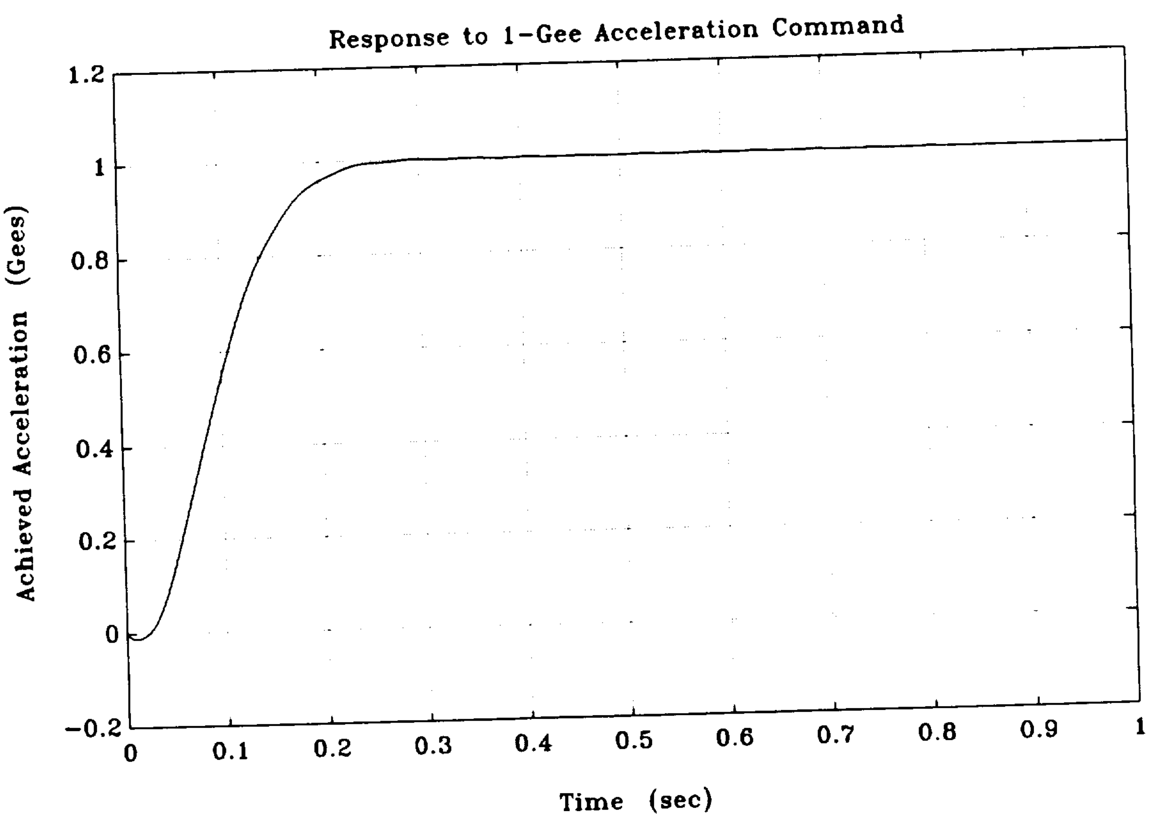
ACCELERATION STEP RESPONSE OF FINAL COMPENSATOR DESIGN



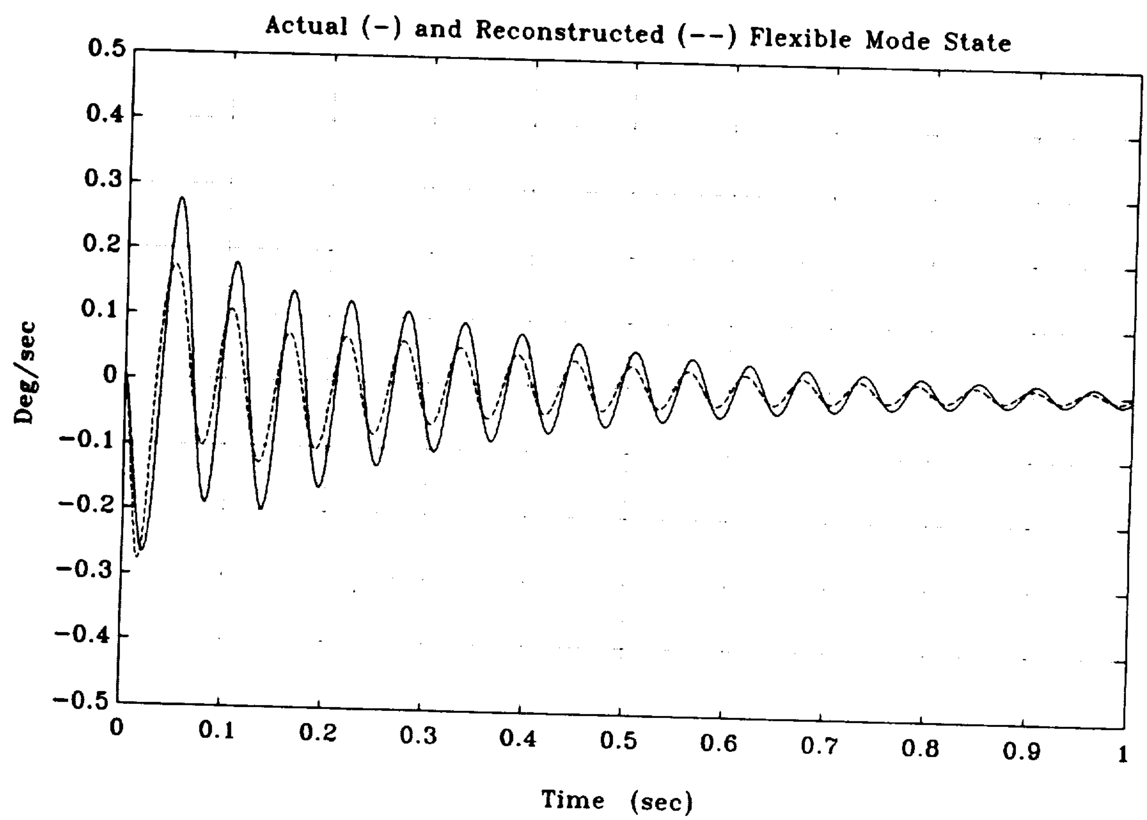
RESPONSE OF FLEXIBLE MODE STATE DURING ACCELERATION STEP RESPONSE



ACCELERATION STEP RESPONSE FOR CASE WHEN BENDING MODE IS PERTURBED TO 25 % LOWER VALUE



COMPARISON OF ACTUAL AND RECONSTRUCTED FLEXIBLE MODE STATE DURING STEP RESPONSE -- BENDING MODE PERTURBED TO 25 % LOWER VALUE



SUMMARY OF DESIGN RESULTS

- Model-based compensator yields a high-bandwidth autopilot, which is robust to at least a 25% perturbation in bending mode frequency

- A number of issues still not addressed:
  - Detailed noise sensitivity assessment
  - Effect of higher-frequency structural modes
  - Phase lag from actuator dynamics
  - Effect of structural modes on accelerometer measurement
  - Tolerance to uncertainties in aerodynamics

- Above concerns could also be addressed using MATLAB

SUMMARY: MATLAB APPLICABILITY FOR CONTROL DESIGN OF FLEXIBLE SYSTEMS

- MATLAB provides the necessary tools for a variety of control system design techniques

- Extensibility of MATLAB allows development of tools to implement recent modern control design methods, including loop transfer recovery

- Implementation for 80386-based machines (PC-MATLAB/386) has very high performance, allowing for interactive control design of complex systems such as flexible structures

- Any flexible structures control problem which can be cast into a state-space framework may benefit from design work with MATLAB