NASA-CR-196112

FINAL REPORT
NASA Langley Research Grant NCC-1-156

Multirate Flutter Suppression System Design for the Benchmark Active Controls Technology Wing

June 16, 1991 - Jun 15, 1994

Martin C. Berg
Associate Professor
Department of Mechanical Engineering, FU-10
University of Washington
Seattle, Washington 98195

Gregory S. Mason
Postdoctoral Research Associate
Department of Mechanical Engineering, FU-10
University of Washington
Seattle, Washington 98195

1. INTRODUCTION

To study the effectiveness of various control system design methodologies, the NASA Langley Research Center initiated the Benchmark Active Controls Project. In this project, the various methodologies will be applied to design a flutter suppression systems for the *Benchmark Active Controls Technology* (BACT)Wing (also called the PAPA wing). Eventually, the designs will be implemented in hardware and tested on the BACT wing in a wind tunnel..

This report describes a project at the University of Washington to design a *multirate* flutter suppression system for the BACT wing. The objective of the project was two fold. First, to develop a methodology for designing robust multirate compensators, and second, to demonstrate the methodology by applying it to the design of a multirate flutter suppression system for the BACT wing.

The contributions of this project are

1) Development of an algorithm for synthesizing robust low order multirate control laws. The algorithm is capable of synthesizing a single compensator which stabilizes both the nominal plant and multiple plant perturbations.

2) Development of a multirate design methodology, and supporting software, for modeling, analyzing and synthesizing multirate compensators.

3) Design of a multirate flutter suppression system for NASA's BACT wing which satisfies the specified design criteria

This report describes each of these contributions in detail. Section 2.0 discusses our design methodology. Section 3.0 details the results of our multirate flutter suppression system design for the BACT wing. Finally, Section 4.0 presents our conclusions and suggestions for future research.

The body of the report focuses primarily on the results The associated theoretical background appears in the three technical papers that are included as Attachments 1-3. Attachment 4 is a user's manual for the software that is key to our design methodology.

CONTENTS

page

2. A METHODOLOGY FOR DESIGNING MULTIRATE COMPENSATORS

2.1. OVERVIEW

Our design methodology defines the general approach a designer would take, and provides the specific tools needed, to solve a multirate control problem. The general approach dictated by the methodology is to model a multirate system as an equivalent single-rate system, to synthesize the compensator using parameter optimization, and to analyze the resulting closed-loop system by applying modified single-rate techniques to a single-rate equivalent model of the multirate system. A schematic of our multirate design methodology is shown in Fig. 2.1. In the following paragraphs we first introduce the terminology and notation unique to multirate systems and then discuss each aspect of the design methodology along with the applicable design and analysis tools.

2.2. DEFINITIONS, ASSUMPTIONS AND NOTATION

A multirate sampled-data system consists of a continuous plant in feedback with a multirate compensator. A block diagram of such a system is shown in Fig. 2.2 where the signals y_S and y_C are continuous output vectors, u is the continuous control input vector, w is the continuous process noise, and v is the discrete sensor noise. The primary components of the multirate system are the continuous plant, the sampling hardware (e.g. A/D converters), a digital processor (e.g., a computer), and the signal holding hardware (e.g., zero-order-hold D/A converters). The samplers, digital processor and holds will be referred to together as the "multirate compensator". We will assume that the plant is linear time-invariant, and that the multirate compensator conforms to the Generalized Multirate Control Law Structure discussed in Section 2.3.1.

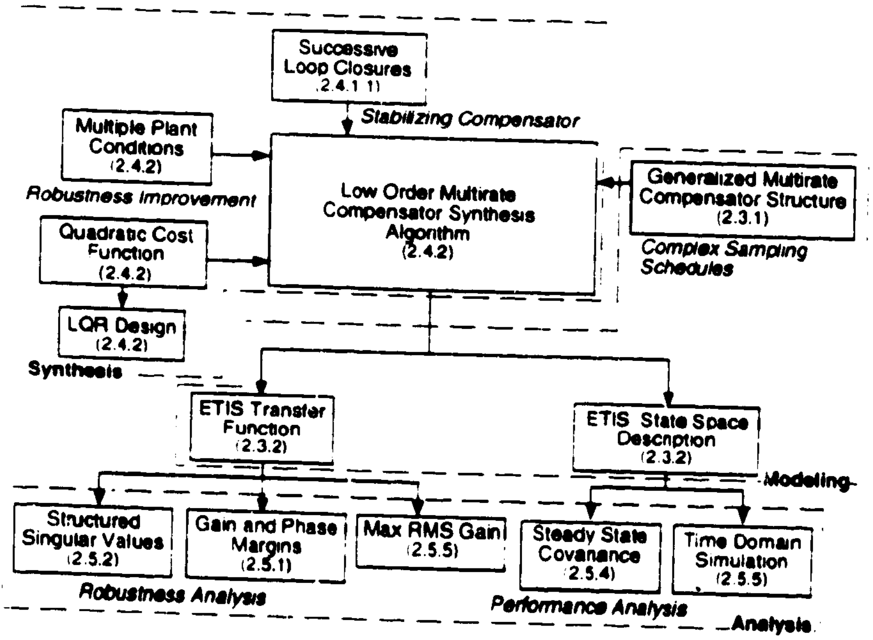

Successive Loop Closures (2.4.1.1)
Stabilizing Compensator
Multiple Plant Conditions (2.4.2)
Robustness Improvement
Low Order Multirate Compensator Synthesis Algorithm (2.4.2)
Generalized Multirate Compensator Structure (2.3.1)
Complex Sampling Schedules
Quadratic Cost Function (2.4.2)
LQR Design (2.4.2)
Synthesis
ETIS Transfer Function (2.3.2)
ETIS State Space Description (2.3.2)
Modeling
Structured Singular Values (2.5.2)
Gain and Phase Margins (2.5.1)
Max RMS Gain (2.5.5)
Steady State Covariance (2.5.4)
Time Domain Simulation (2.5.5)
Robustness Analysis
Performance Analysis
Analysis

Figure 2.1. A multirate design methodology. Section numbers indicated in parentheses

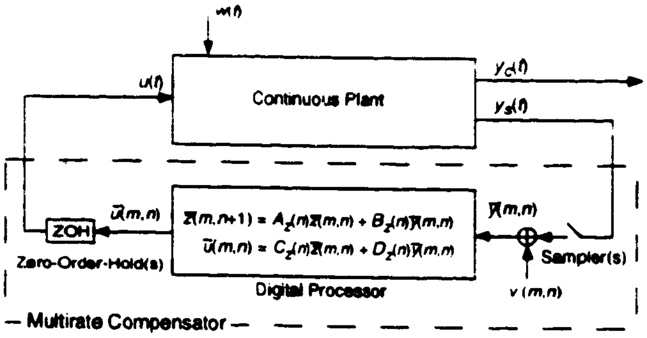

w(t)
u(t)
Continuous Plant
$y_c(t)$
$y_s(t)$
$\bar{u}(m,n)$
ZOH
Zero-Order-Hold(s)
$\bar{z}(m,n+1) = A_z(n)\bar{z}(m,n) + B_z(n)\bar{y}(m,n)$
$\bar{u}(m,n) = C_z(n)\bar{z}(m,n) + D_z(n)\bar{y}(m,n)$
Digital Processor
$\bar{y}(m,n)$
Sampler(s)
v (m,n)
— Multirate Compensator —

Figure 2.2. Multirate sampled-data system

As we will later see, multirate systems which satisfy our assumptions are periodically time-varying. To emphasize their periodic nature we will use a double index notation for the independent variable of a sampled or discrete signal. For example, given a continuous signal $y(t)$, $y(m,n)$ represents $y(t)$ sampled at the time $t=(mN+n)T$; where the integer N is the period of repetition; T is the sampling period; m =0, 1, ... ; and n=0, 1, ... N-1.

The design methodology presented in the following sections provides tools to model the closed loop system in Fig. 2.2, to compute optimum values of A_z, B_z, C_z and D_z, and to analyze the performance of the closed-loop system.

2.3. MODELING A MULTIRATE SYSTEM

Two useful modeling tools are the Generalized Multirate Control Law Structure (GMCLS) and the Equivalent Time-Invariant System (ETIS).

2.3.1. The GMCLS

The GMCLS is a control law structure which describes a multirate compensator of arbitrary dynamic order, with an independent sampling rate for every compensator input, and independent update rates for every processor state and compensator output. A multirate compensator with the GMCLS is shown in Fig. 2.2. In this figure each element of the continuous plant output y_s is sampled at an independent rate. The sampled value of y_s, $\bar{y}$, is combined with the current processor state vector, $\bar{z}$, using the state space structure shown in the figure. Each element of the processor state vector, $\bar{z}$, is updated at an independent rate. The continuous output from the compensator, represented by the vector u, is formed by holding the output from the digital processor, $\bar{u}$, with a zero-order-hold. Each element of the vector $\bar{u}$ can be held at an independent rate to form u.

Conceptually, one can divide the multirate compensator into two parts, the "sampling schedule" and the digital processor gains. This is the approach used in the GMCLS. The "sampling schedule" is a description of when each compensator input is sampled and when each compensator output and processor state is updated, while the digital processor gains determine the dynamics of the digital processor.

2.3.1.1. *Sampling Schedule for a GMCLS*

In general, the sampling and updating of the elements of y_s, $\bar{z}$, and $\bar{u}$ in Fig. 2.2 can occur at any time. However, to conform to the GMCLS, we require that these sample and update activities occur only at integer multiples of some fixed time, called the ***shortest time period*** (STP). The actual value of the STP is arbitrary, but it is often a function of the hardware and software used to implement the control law. We also require that the sampling and updating activities of the sensors, states and outputs repeat themselves after some fixed period of

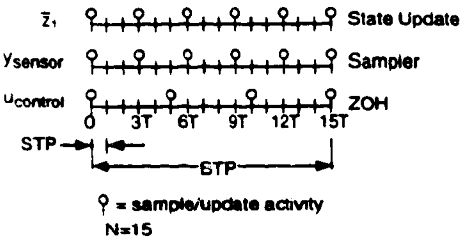


Figure 2.3 Example Sampling Schedule

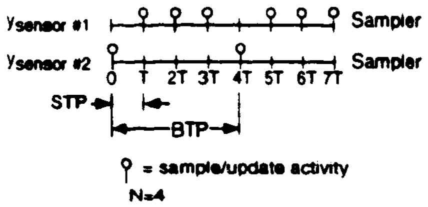


Figure 2.4 Aperiodic Sampling Schedule

time. (This requirement disallows, for example, a system whose sampling period is a function of the time require to execute the control software which might vary with control inputs values.) The period of repetition of the sampling schedule is called the *basic time period* (BTP). Finally, we define

$$\text{the integer } N = \frac{\text{BTP}}{\text{STP}} \quad \text{and the value} \quad T = \text{STP} \tag{2.1}$$

In our double index notation, the first index (*m*) in, for example, *y(m,n)* indicates the integer number of BTP's which have elapsed when the sample/update occurred and the second index (*n*) indicates the integer number of STP's which have elapsed within the current BTP when the sample/update occurred.

We can represent the sampling schedule for the multirate compensator graphically, as shown in Fig. 2.3. The figure shows a time line for each sampler, processor state, and zero-order-hold. The time line is divided into one STP increments. On the left side of the time line is a description of the signal or state being sampled or updated. On the right side is a description of the particular activity represented by the time line, e.g., state update, sampler, or zero-order-hold. Circles on each time line indicate when a sample or update activity associated with that particular signal or state takes place. Usually the sampling schedule is shown for only one BTP since the sampling schedule repeats itself every BTP.

In most applications, the sampling/updating activities for a given sensor, output or state will be periodic *within* the BTP, as is shown in Fig. 2.3. However, the sampling/updating activities do not have to be periodic within the BTP. The only requirement is that the sampling/updating activities have some period of repetition (the BTP) and that they occur at integer multiples of the STP. Once the STP and BTP have been selected, the designer can arbitrarily specify sampling/updating activities at any multiple of the STP within one BTP. An example of a multirate sampling schedule in which the sampling/updating activities are not periodic within the BTP is shown in Fig. 2.4. A sampling policy like this might be used to multiplex multiple inputs through a single analog to digital converter.

2.3.1.2. Digital Processor Gains

The processor gains are the values of the matrices A_z, B_z, C_z, and D_z in Fig. 2.2. Like the sampling schedule, they can be periodically time-varying with a period of repetition of one BTP. Generally, these matrices are free design parameters which can be adjusted by the designer to improve the performance of the multirate compensator. The synthesis algorithm discussed in Section 2.4 can be used to calculate optimum values for these gains.

2.3.1.3. *State Space Formulation of the GMCLS*

A compensator with the GMCLS can be modeled as a periodically time-varying discrete-time system. The state space form of the GMCLS is given by

$$z(m,n+1) = A_g(n)z(m,n) + B_g(n)y(m,n) \tag{2.2a}$$

$$u(m,n) = C_g(n)z(m,n) + D_g(n)z(m,n) \tag{2.2b}$$

where

$$z(m,n) = [\bar{z}(m,n)^{\mathrm{T}} \quad \bar{y}(m,n)^{\mathrm{T}} \quad \tilde{u}(m,n)^{\mathrm{T}}]^{\mathrm{T}} \tag{2.3}$$

and $\tilde{u}(m,n)$ is used to model the sample and hold activity from $\bar{u}(m,n)$ to $u(m,n)$. The form of A_g, B_g, C_g and D_g is given in [Berg, Mason & Yang 1991] and [Mason & Berg 1992] which are included as Attachments 1 and 2.

We should emphasize that Eqn. (2.2) is used to model the *complete* sampling/updating activities and dynamics of a multirate compensator. It would not be used in the actual implementation of the compensator. When implemented, the sample and hold activities of the inputs and outputs would be performed by appropriate hardware. The only dynamics to be calculated are those associated with the processor state vector $\bar{z}$.

2.3.1.4. *Factored Form of the GMCLS*

Equation (2.2) is a convenient form to model the general multirate compensator. The difficulty with Eqn. (2.2) is that it ties up the digital processor matrices, $A_z(n)$, $B_z(n)$, $C_z(n)$, and $D_z(n)$, in the model matrices $A_g(n)$, $B_g(n)$, $C_g(n)$, and $D_g(n)$. The matrices $A_z(n)$, $B_z(n)$, $C_z(n)$, and $D_z(n)$, which describe the dynamics of the digital processor, are the unknown design parameters which we will later optimize. We can separate the processor dynamics matrices from the model matrices as follows.

Define the composite compensator matrix:

$$P(n) = \begin{bmatrix} D_g(n) & C_g(n) \\ B_g(n) & A_g(n) \end{bmatrix} \tag{2.4}$$

and factor $P(n)$ as follows

$$P(n) = S_1(n)P_z(n)S_2(n) + S_3(n) \tag{2.5}$$

$$\text{where } P_z(n) = \begin{bmatrix} D_z(n) & C_z(n) \\ B_z(n) & A_z(n) \end{bmatrix} \tag{2.6}$$

and S_1, S_2 and S_3 are the switching matrices defined by the sampling schedule for the compensator. Their exact form is given in [Mason 1992] and [Mason & Berg 1992] (see Attachment 1).

It is important to note the difference between $P(n)$ and $P_z(n)$ in Eqn. (2.5). $P(n)$ is a periodically time-varying matrix defined by Eqn. (2.4). It includes all the information about the processor gains *and* the sampling/update schedule. $P_z(n)$ contains *only* the gains for the processor dynamics and is independent of the sampling schedule.

2.3.1.5. *Implementation*

The Generalized Multirate Control Law Structure (GMCLS) provides a framework for dealing with multiple sample/update rates, time delays, and periodically time-varying gains in a digital control system. It gives the designer freedom to either select the "sampling schedule" that best solves the problem, or if necessary, to use the "sampling schedule" dictated by existing hardware and software, with out having to worry about the bookkeeping involved with multiple rates and time delays.

Ir practice, the GMCLS is implemented in software and is rarely used directly by the designer. The designer need only supply the sampling schedule and values for the digital processor gains to provide a complete compensator description. This description can then be transformed directly into a single-rate periodically time-varying system using the GMCLS.

The GMCLS is used extensively by the synthesis algorithm described in Section 2.4, and by the modeling and analysis software referred to in Section 2.5. Documentation for this software is provided in Attachment 4

2.3.2. The Equivalent Time-Invariant System (ETIS)

A multirate compensator with the periodically time-varying structure discussed in Section 2.3.1.3 can be further transformed into a single-rate *Equivalent Time-Invariant System* (ETIS) with the form shown below

$$x(m+1,0) = A_E\, x(m,0) + B_E u_E(m,0) \tag{2.7a}$$

$$y_E(m,0) = C_E\, x(m,0) + D_E u_E(m,0) \tag{2.7b}$$

where

$$y_E(m,0) = \begin{bmatrix} y_s(m,0) \\ y_s(m,1) \\ \vdots \\ y_s(m,N-1) \end{bmatrix} \text{ and } u_E(m,0) = \begin{bmatrix} u(m,0) \\ u(m,1) \\ \vdots \\ u(m,N-1) \end{bmatrix} \tag{2.8}$$

We use the subscript E to denote vectors and matrices strictly associated with the ETIS. See [Meyer & Burrus 1975] or [Mason 1992] for a definition of A_E, B_E, C_E and D_E.

A key feature of an ETIS is that a multirate, or periodically time-varying system will be stable if and only if its ETIS is stable [Kono 1971]. Also notice that the ETIS input/output vectors are composite vectors containing the input/output values of the multirate (or periodically-time varying) system at N sampling times. Consequently, an ETIS is always MIMO even if the original system is SISO. If the multirate system has p inputs, q outputs and a sampling period of one STP then the ETIS is a single-rate linear time-invariant system with $N\,p$ inputs, $N\,q$ outputs and a sampling period of one BTP.

2.3.2.1. Implementation

The ETIS is fundamental to the analysis of multirate systems. It allows one to evaluate the performance and stability of complex systems comprised of multirate, periodically time-varying and/or single-rate components using only techniques developed for linear time-invariant single-rate systems. For example, to evaluate the stability of the system in Fig. 2.2, we would first transform the multirate compensator into its ETIS with a given value for N. Then we would discretize the plant at the STP of the compensator using a zero-order-hold and transform the resulting single-rate system into an ETIS using the BTP of the compensator. Next, the plant and compensator ETIS's could be combined in feedback just as if they were traditional single-rate systems. Finally, we could determine the stability of the original multirate sampled-data system from the eigenvalues of its closed-loop ETIS.

Documentation for software capable of transforming multirate and single-rate systems into their ETIS's is provided in Attachment 4.

2.4. SYNTHESIZING A MULTIRATE COMPENSATOR

When designing a multirate compensator for the system in Fig. 2.2 there are three components one must consider: the compensator structure (this includes the dynamical order of the digital processor), the sampling schedule, and the values for the digital processor gains. In our design methodology the compensator structure and sampling schedule are selected by the designer based on the open-loop plant dynamics, the hardware constraints, if any, and the desired closed-loop performance. Values for the digital processor gains are calculated by our synthesis algorithm so as to provide optimum closed-loop performance for the chosen compensator structure and sampling schedule. In the following paragraphs we discuss compensator structure and sampling schedule selection, and provide a brief description of our synthesis algorithm. A complete discussion of the algorithm is provided in Attachment 4 and in [Mason & Berg 1992] (also included as Attachment 1).

2.4.1. Compensator Structure and Sampling Schedule Selection

The choice of compensator structure and sampling schedule is problem dependent. It depends on the hardware constraints, the open-loop plant dynamics, and the design objectives. Two often used multirate compensator structures are worthy of mention, however. They are successive loop closure and coupled successive loop closures. (Also see [Berg 1986] for a discussion of successive loop closures.)

2.4.1.1. Successive Loop Closures Structure

The simplest multirate compensator structure is successive loop closures (SLC). This structure consists of multiple decoupled single-rate control loops, each loop operating at a unique sample/update rate. The state space representation of a SLC structure with two loops is

$$\begin{Bmatrix} x_{fast}(m+1) \\ x_{slow}(n+1) \end{Bmatrix} = \begin{bmatrix} a_{fast} & 0 \\ 0 & a_{slow} \end{bmatrix} \begin{Bmatrix} x_{fast}(m) \\ x_{slow}(n) \end{Bmatrix} + \begin{bmatrix} b_{fast} & 0 \\ 0 & b_{slow} \end{bmatrix} \begin{Bmatrix} y_{fast}(m) \\ y_{slow}(n) \end{Bmatrix} \tag{2.9a}$$

$$\begin{Bmatrix} u_{fast}(m) \\ u_{slow}(n) \end{Bmatrix} = \begin{bmatrix} c_{fast} & 0 \\ 0 & c_{slow} \end{bmatrix} \begin{Bmatrix} x_{fast}(m) \\ x_{slow}(n) \end{Bmatrix} + \begin{bmatrix} d_{fast} & 0 \\ 0 & d_{slow} \end{bmatrix} \begin{Bmatrix} y_{fast}(m) \\ y_{slow}(n) \end{Bmatrix} \tag{2.9b}$$

where y represents the sampled input from the sensor and u is the output to the zero-order-hold. The subscripts *fast* and *slow* denote inputs, outputs and states which are sampled/updated at a fast or slow rate, respectively.

SLC is best applied to control problems where the *closed-loop* dynamics are comprised of some fast and some slow dynamics with the bandwidths of the two separated by at least a factor of four. In this type of problem, the "fast" loop(s) of the SLC compensator, operating at a fast sampling/update rate, would be used to control the high bandwidth dynamics, while the "slow" loop(s), operating at a slower sampling/update rate, would be used to control the low bandwidth dynamics. Problems such as these usually fall into one of two categories.

In the first, the *open-loop* system exhibits both fast and slow dynamics. The multirate compensator is used to improve the performance of this system without drastically changing the fast or slow bandwidths. An example of this type of problem is an aircraft yaw damper/modal suppression system. The aircraft is open-loop stable and has some fast dynamics associated with the flexibility of the airframe and some slower dynamics associated with the yawing motion of the entire aircraft. A multirate compensator for such a system might consist of a high bandwidth loop to damp the airframe vibrations and a low bandwidth loop to improve yaw damping.

In the second type, the open-loop dynamics of the plant are arbitrary, but in feedback with the compensator the closed-loop system exhibits the characteristic fast and slow dynamics. These systems usually have a decoupled structure where sets of open-loop modes are strongly controllable and observable with a particular set of inputs and outputs and weakly controllable and observable with the remaining inputs and outputs. An example of this type of system is the two link robot arm (TLA) used in [Berg, Amit & Powell 1988], and in [Yang 1988]. All four of the open-loop poles of the TLA are at the origin of the "s" plane. The plant has two inputs and two outputs. Only two of the modes can be controlled with any one input. Similarly, only two of these modes can be observed with any one output. In the multirate design, one input/output pair is used to place two of the closed-loop poles at a high frequency and the other input/output pair is used to place the other two closed-loop poles at a low frequency.

Sample rate selection for the individual control loops of a SLC design follows the same guide lines used in single-rate sample rate compensator design: the sample rate for each SLC loop should be 5 to 20 times faster than the closed-loop bandwidth desired for that loop. See [Franklin Powell & Workman 1990] for a discussion of sample rate selection for single-rate systems.

2.4.1.2. Coupled Successive Loop Closures Structure

The coupled SLC structure is the same as the traditional SLC structure except the designer can include cross feed terms which couple the fast and slow inputs and outputs of the design. In the state space formulation, cross coupling is represented by non-zero off diagonal terms in the compensator gain matrices. An example of a compensator structure with cross feed from the slow sampled sensor to the fast sampled/updated control loop is given in Eqn. (2.10).

$$\begin{Bmatrix} x_{fast}(m+1) \\ x_{slow}(n+1) \end{Bmatrix} = \begin{bmatrix} a_{fast} & 0 \\ 0 & a_{slow} \end{bmatrix} \begin{Bmatrix} x_{fast}(m) \\ x_{slow}(n) \end{Bmatrix} + \begin{bmatrix} b_{fast} & b_{fs} \\ 0 & b_{slow} \end{bmatrix} \begin{Bmatrix} y_{fast}(m) \\ y_{slow}(n) \end{Bmatrix} \tag{2.10a}$$

$$\begin{Bmatrix} u_{fast}(m) \\ u_{slow}(n) \end{Bmatrix} = \begin{bmatrix} c_{fast} & 0 \\ 0 & c_{slow} \end{bmatrix} \begin{Bmatrix} x_{fast}(m) \\ x_{slow}(n) \end{Bmatrix} + \begin{bmatrix} d_{fast} & d_{fs} \\ 0 & d_{slow} \end{bmatrix} \begin{Bmatrix} y_{fast}(m) \\ y_{slow}(n) \end{Bmatrix} \tag{2.10b}$$

This structure is best applied to systems which have coupling between their fast and slow closed-loop dynamics. See [Yang 1988] for a discussion of cross feed in the TLA problem.

2.4.2. Optimizing the Digital Processor Gains

Having chosen an appropriate compensator structure and sampling schedule, the designer can use our synthesis algorithm to calculate optimum values for the digital processor gains A_z, B_z, C_z and D_z such that the closed-loop system in Fig. 2.2 minimizes a quadratic cost function.

The primary design parameter for the synthesis algorithm is the quadratic cost function. By selecting an appropriate cost function, the designer can influence the performance of the resulting closed-loop system. The cost function minimized by our synthesis algorithm has the form:

$$\bar{J} = \lim_{t\to\infty} E\left\{ \begin{bmatrix} y_c^T(t) & u^T(t) \end{bmatrix} \begin{bmatrix} Q_1 & M \\ M^T & Q_2 \end{bmatrix} \begin{bmatrix} y_c(t) \\ u(t) \end{bmatrix} \right\} \tag{2.11}$$

where $\bar{J}$ is the cost associated with the closed-loop system shown in Fig. 2.2. The vector y_c is the continuous criterion output and u is the continuous control input. Q_1, Q_2 and M are the cost function weighting matrices and are free design parameters.

The cost function in Eqn. (2.11) has the same form in a continuous time LQR design. Thus the cost associated with the optimized multirate compensator and that of an LQR design can be compared directly. The designer can also use this fact to help select appropriate values for Q_1, Q_2 and M.

To improve the robustness of the compensator, the synthesis algorithm can optimize the digital processor gains for multiple plant conditions simultaneously. The resulting compensator will stabilize the each plant condition and provide overall optimum performance. This is accomplished by minimizing the new cost function of Eqn. (2.12) which is the sum of the costs associated with each plant condition.

$$J = \sum_{i=1}^{\mathbf{Np}} \bar{J}_i = \sum_{i=1}^{\mathbf{Np}} \lim_{t\to\infty} E\left\{ \begin{bmatrix} y_{ci}^{\mathrm{T}}(t) & u_i^{\mathrm{T}}(t) \end{bmatrix} \begin{bmatrix} Q_i & M_i \\ M_i^{\mathrm{T}} & R_i \end{bmatrix} \begin{bmatrix} y_{ci}(t) \\ u_i(t) \end{bmatrix} \right\} \tag{2.12}$$

Here $\bar{J}_i$ is the cost associated with the i^{th} plant perturbation and there are Np plant perturbations.

Optimum values of A_z, B_z, C_z, and D_z, occur when

$$\frac{\partial J}{\partial A_z} = 0,\ \frac{\partial J}{\partial B_z} = 0,\ \frac{\partial J}{\partial C_z} = 0, \text{ and } \frac{\partial J}{\partial D_z} = 0 \text{ or equivalently when } \frac{\partial J}{\partial P_z} = 0 \tag{2.13}$$

Our algorithm use a gradient type numerical search and a closed form expression for the gradients in Eqn. (2.13) to determine values of the digital processor gains such that the conditions in Eqn. (2.13) are satisfied. Refer to [Mason & Berg 1992] in Attachment 1 for a closed form expression for the gradients in Eqn. 2.13. The synthesis software uses an iterative process to determine optimum values for the digital processor gains and the user must provide the software with an initial guess for A_z, B_z, C_z, and D_z. The initial guess must stabilize every plant condition considered in Eqn. (2.12).

2.4.3. Implementation

In practice, the steps for designing a compensator with our methodology are

1) Construct a continuous LQ regulator for each plant condition which achieves the desired performance for that condition.

2) Based on the desired closed-loop dynamics and the constraints imposed by the system hardware, choose an appropriate compensator structure and sampling schedule.

3) Using the chosen sampling schedule and compensator structure, design a compensator which stabilizes all plant perturbations. When the desired compensator structure is one of the two structures discussed in the previous section, the designer can use successive loop closures to find a stabilizing value for the digital processor gains. In successive loop closures, the plant is stabilized by closing one loop at a time, from one set of inputs to one set of outputs. To obtain a multirate compensator, each loop is closed using a different sampling/update rate. When, due either to a complex sampling schedule, or the complexities of the control problem, successive loop closures cannot be used to find a stabilizing value for the digital processor gains, use Yang's algorithm (see [Yang 1988]). This may seem counterproductive at first, since one of the reasons for developing our algorithm was the computational inefficiencies of Yang's algorithm. However, our experience has shown that, in general, Yang's algorithm converges to a stabilizing compensator fairly rapidly. It is the computation time associated with optimization of this stabilizing solution that tends to be excessive.

4) Calculate optimum values for the digital processor gains using the synthesis algorithm of Section 2.4.2. The cost function weighting matrices for the optimization are the same as those used to design the LQ regulators in Step 1. The starting point for the optimization is the stabilizing compensator designed in Step 3.

See Attachment 4 for the complete documentation of the software that implements the synthesis algorithm.

2.5. ANALYZING A MULTIRATE SYSTEM

Multirate system analysis is difficult because the periodic nature of a multirate system implies that a traditional transfer function does not exist. Thus, common analysis tools such as frequency response or Nyquist diagrams are not directly applicable to multirate systems. Our solution is to transform the multirate system into a linear time-invariant single-rate system, the ETIS, and then apply established single-rate analysis techniques using the Z-Transform of the ETIS. (Note: we write the Z-Transform of an ETIS where N=BTP/STP as $G_E(z^N)$.) The following paragraphs discuss five useful tools for analyzing the performance and stability of a multirate system based on its ETIS.

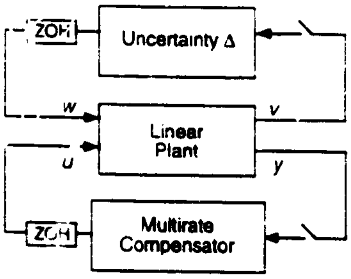


Figure 2.5. Plant/compensator configuration

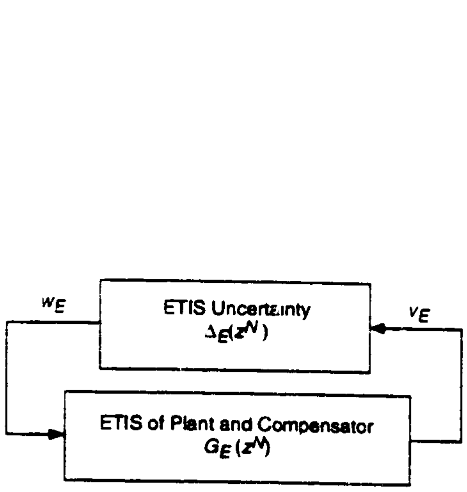

w_E
ETIS Uncertainty $\Delta_E(z^N)$
v_E
ETIS of Plant and Compensator $G_E(z^N)$

Figure 2.6. Plant/compensator with uncertainty

2.5.1. Gain and Phase Margins

In Section 2.3.2 we noted that a multirate system will be stable if and only if its ETIS is stable. Therefore, we can determine whether the multirate system is stable by applying the Nyquist criterion to its ETIS. Since all but trivial ETIS's are MIMO, we must use the multiloop Nyquist stability criterion. The multivariable Nyquist is a plot of the eigenvalues of the ETIS loop transfer function as the discrete variable z traverses the unit circle [MacFarlane 1970] [Maciejowske 1990]

When the multirate system is SISO we can obtain traditional gain and phase margins from the multiloop Nyquist plot. Let $G_E(z^N)$ be the ETIS loop transfer function and let Δ be some constant gain and phase uncertainty at the plant input. If

$$\Delta(z) = k e^{j\theta} \text{ where } k e^{j\theta} \text{ is a scalar} \tag{2.14}$$

$$\text{then } \Delta_E(z^N) = I k e^{j\theta} \tag{2.15}$$

where I is and $N \times N$ identity matrix

Now the new loop transfer function with the gain and phase uncertainty of Eqn. (2.15) can be written as

$$H_E(z^N)_{\text{loop}} = G_E(z^N)\, k e^{j\theta} \tag{2.16}$$

The multiloop Nyquist plot of $H_E(z^N)_{\text{loop}}$ is just the multiloop Nyquist plot of $G_E(z^N)$ scaled by the gain k and rotated by the phase shift θ - the same as in traditional SISO Nyquist plots. Gain and phase margins for the multirate system can therefore be obtained from the multiloop Nyquist plot of $G_E(z^N)$ by determining the values of k and θ which destabilize the ETIS. (See [Thompson 1986] for an alternate derivation using Kranc operators.)

When the multirate system is MIMO, the gain and phase margins calculated by this procedure apply simultaneously to all inputs and outputs, and are consequently not realistic measures of robustness. To obtain realistic measures of robustness for a MIMO multirate system, a norm based approach such as singular value analysis is required.

2.5.2. Singular Values

Singular values are useful for measuring the robustness of MIMO multirate systems. The key step in multirate singular value analysis is transforming the multirate system in Fig. 2.5 into an ETIS system which has the output feedback form shown in Fig. 2.6. Since the multirate system will be stable if and only if its ETIS is stable, the closed-loop system in Fig. 2.5 will be stable for a given value of Δ provided the closed-loop system in Fig. 2.6 is stable for a corresponding value of Δ_E. Thus we can use single-rate techniques to evaluate the robustness of the ETIS system and relate those results directly to the associated multirate system.

2.5.2.1. *Unstructured Singular Value Analysis*

A bound on the smallest value of $\bar{\sigma}(\Delta_E)$ for which Δ_E destabilizes the system shown in Fig. 2.6 can be calculated using unstructured singular value analysis. This system will be stable for all Δ_E such that

$$\bar{\sigma}(\Delta_E(z^N)) < \frac{1}{\bar{\sigma}(G_E(z^N))} \quad \text{for all } z \text{ on the unit circle} \tag{2.17}$$

(see [Maciejowski 1989]). This result, however, is only a measure of the size of the smallest destabilizing Δ_E and is generally not a measure of the size of the smallest destabilizing uncertainty Δ. Because the input/output vectors of an ETIS are composite vectors, containing the input/output values of the multirate system at N sample times, Δ_E can be a complex function of the values of Δ at N sample times. (The relation between Δ_E and Δ is given by Eqn. 2.7.) The size of the smallest destabilizing Δ_E found using unstructured singular value analysis is only a conservative estimate of the size of the smallest destabilizing Δ. This estimate accounts for not only the fictitious perturbations normally associated with unstructured singular values, but also for time-varying and non-causal perturbations.

Consider the simple case where Δ is a constant. From Fig. 2.5 we have that

$$w = \Delta v \tag{2.18}$$

For an ETIS with N=2

$$w_E = \Delta_E v_E \text{ or } \begin{Bmatrix} w(m,0) \\ w(m,1) \end{Bmatrix} = \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix} \begin{Bmatrix} v(m,0) \\ v(m,1) \end{Bmatrix} \tag{2.19}$$

A destabilizing Δ_E determined by singular value analysis might, for example, include block diagonal elements in Δ_E which are unequal, e.g. $\Delta_{11} \neq \Delta_{22}$. This corresponds to a time-varying perturbation because the gain between w and v varies with time. Another such Δ_E could include non-zero upper block diagonal elements in Δ_E, e.g. $\Delta_{12} \neq 0$. This corresponds to a non-causal perturbation because a future input, $v(m,1)$, can affect the current output $w(m,0)$.

We can eliminate this conservativeness by restricting the allowable perturbations in Δ_E. This leads directly to structured singular value analysis.

2.5.2.2. *Structured Singular Value Analysis*

In order for the ETIS uncertainty Δ_E to represent the actual uncertainty Δ, its structure must obey Eqn. (2.7). Finding the size of the smallest destabilizing Δ_E subject to Eqn. (2.7) requires the solution of a structured singular value problem. For the system in Fig. 2.6 we define the structured singular value, μ, as

$$\mu(G_E(z^N)) = \begin{cases} 0 \text{ if } \det(I - G_E(z^N)\Delta_E(z^N)) \neq 0 \text{ for all } \Delta \in \Delta_{BD} \\ \left(\min\limits_{\Delta \in \Delta_{BD}} [\bar{\sigma}(\Delta(z))] \text{ such that } \det(I - G_E(z^N)\Delta_E(z^N)) = 0 \right)^{-1} \text{ otherwise} \end{cases} \tag{2.20}$$

where Δ_{BD} is the form of the permissible block diagonal perturbations Δ and the structure of Δ_E must satisfy Eqn. (2.7). The size of the smallest destabilizing perturbation $\bar{\sigma}(\Delta_{\min})$ satisfies

$$\frac{1}{\bar{\sigma}(\Delta_{\min})} = \sup_{\phi} \mu(G_E(z^N)) \quad \text{where } z^N = e^{j\phi} \tag{2.21}$$

For a discussion of μ and Δ_{BD} see [Doyle 1982].

Unfortunately, even a simply structured dynamic uncertainty $\Delta(z)$ transforms to an ETIS uncertainty, $\Delta_E(z^N)$, with a complex structure. For example, if N=2 then the ETIS of $\Delta(z)$ is

$$\Delta_E(z^2) = \frac{1}{2}\begin{bmatrix} \Delta(z)+\Delta(-z) & z^{-1}(\Delta(z)-\Delta(-z)) \\ z(\Delta(z)-\Delta(-z)) & \Delta(z)+\Delta(-z) \end{bmatrix} \tag{2.22}$$

In order to find the value of $\bar{\sigma}(\Delta_{\min})$ using Eqn. (2.21), one must solve Eqn. (2.20) with Δ_E constrained to have the structure in Eqn. (2.7). Currently there is no general technique for solving this problem. When, however, the uncertainty, Δ, is a constant, as is the case for many problems, the ETIS uncertainty, Δ_E, is also a constant with a repeated block diagonal form.

$$\Delta_E = diag(\Delta, \Delta, \ldots, \Delta) \text{ with } N \text{ blocks.} \tag{2.23}$$

There are several good methods for estimating $\bar{\sigma}(\Delta_{\min})$ when Δ_E has this block diagonal structure. One simple method for estimating μ when Δ is *strictly* diagonal is derived in [Safonov 1982]. It is

$$\mu(G_N(z^N)) \leq \inf_D(\bar{\sigma}(\text{abs}(DG_N(z^N)D^{-1})) = \lambda_p(G_N(z^N)) \tag{2.24}$$

where $\text{abs}(A)$ is a matrix such that $[\text{abs}(A)]_{ij} = |A_{ij}|$; A_{ij} is the $i^t j^{th}$ element of A; and λ_p is the Perron-Frobenius eigenvalue.

2.5.4.3 Implementation

The procedure for performing singular value analysis via the ETIS is as follows

1) Transform the problem into the form shown in Fig. 2.5

2) Discretize the plant at the STP of the compensator and compute the ETIS of the plant using the N of the compensator

3) Combine the ETIS of the plant and compensator to obtain the closed-loop system shown in Fig. 2.6

4) Use any applicable single-rate singular-value based analysis tool to compute the size of the smallest destabilizing uncertainty Δ_E.

5) Interpret the results in the light of the fact that the computed results are for an ETIS uncertainty Δ_E whereas the actual plant uncertainty is Δ. Δ_E is a function of Δ as given by Eqn. 2.7 and so the results might be conservative unless structured singular value analysis is used.

2.5.3. Maximum RMS Gain

The maximum RMS gain of a SISO *single-rate* system is the maximum gain on that system's Bode plot. As already noted, a traditional Bode plot cannot be generated for a *multirate* system. However, the maximum RMS gain of a SISO *multirate* system can be computed; it is the H_∞ norm of the ETIS transfer function. This value, shown in Eqn. (2.25), plays the same role as the maximum Bode plot gain of a *single-rate* system.

$$\sup_{\text{RMS}(u)\neq 0} \frac{\text{RMS}(y(m,n))}{\text{RMS}(u(m,n))} = \sup_{\text{RMS}(u_E)\neq 0} \frac{\text{RMS}(y_E(m,0))}{\text{RMS}(u_E(m,0))} = \|G_E(z^N)\|_\infty \tag{2.25}$$

Actually, Eqn. (2.25) can be used to calculate the RMS gain of SISO or MIMO systems. It simply states that the maximum RMS gain of a transfer function G_E is equivalent to the H_∞ norm of G_E. See also the related work of [Sivashankar & Khargonekar 1991].

Unlike linear time invariant single-rate systems, the discrete input signal resulting in the maximum multirate RMS gain does not necessarily have the simple form $\sin(\omega T m)$. Instead it is comprised of the sum of sinusoids of several distinct frequencies. Details on computing the signal of maximum RMS gain for a multirate system are given in [Mason & Berg 1992] (Attachment 1).

2.5.3.1 Implementation

One simple method for determining the H_∞ norm is to plot the maximum singular value of G_E as z traverses the unit circle. $H_\infty(G_E)$ is then the peak value on that plot.

It is important to remember that Eqn. (2.25) is a measure of the *discrete* RMS gain between the *discrete* inputs and outputs of interest. Often the designer is interested in calculating the maximum RMS gain between a continuous input and output of a sampled-data system. A good estimate of the RMS gain in this case can be found by sampling the continuous input and output of interest at a fast rate. The result is a multirate system - the input and output of interest are sampled/updated at a fast rate while the other inputs and outputs are sampled at the rate appropriate for connection to the multirate compensator. (This is also useful for determining the inter-sample behavior of a sampled-data system.) The maximum RMS gain can then be calculated using the ETIS of this new system.

2.5.4. Steady-State Covariance

A common measure of performance is the steady-state covariance of select outputs in response to a disturbance input. In a multirate system the "steady-state" covariance values are periodically time-varying. Fortunately, the periodic "steady-state" covariance values at each sample/update time are straightforward to calculate using the ETIS.

It is easy to show that

$$E\{y_E y_E^T\} = \begin{bmatrix} E\{y(m,0)y(m,0)^T\} & E\{y(m,0)y(m,1)^T\} & \cdots & E\{y(m,0)y(m,N-1)^T\} \\ E\{y(m,1)y(m,0)^T\} & E\{y(m,1)y(m,1)^T\} & & E\{y(m,1)y(m,N-1)^T\} \\ \vdots & \vdots & \ddots & \vdots \\ E\{y(m,N-1)y(m,0)^T\} & E\{y(m,N-1)y(m,1)^T\} & \cdots & E\{y(m,N-1)y(m,N-1)^T\} \end{bmatrix} \tag{2.26}$$

The diagonal block elements of Eqn. (2.26) contain the steady-state covariance values at each sample/update time of the corresponding multirate system. Therefore, the steady-state covariance values can be found by calculating the ETIS of the multirate system and computing the steady-state covariance values of the ETIS using the discrete Lyapunov equation. Refer to [Kwakernaak & Sivan 1972]. Algorithms for calculating discrete covariance values are widely available (e.g., in Matlab and in Matrix_X).

2.5.5. Time Domain Simulations

Time domain simulations are straightforward to compute using the ETIS and Eqn. (2.7). As noted in Section 2.5.3, inter-sample behavior can be obtained by sampling the continuous inputs and outputs at an arbitrarily fast rate. Documentation for the M-File **mrsim**, which generates a time domain simulation of a multirate sampled-data system using the ETIS is provided in Attachment 4.

2.6. SUMMARY

The tools presented in this section form the foundation of our multirate design methodology, and provide a unified approach to multirate modeling, synthesis and analysis. Using these tools one can model a complex multirate compensator, determine the optimum values of that compensator's processor gains, and analyze its robustness and performance. In many cases the multirate systems modeling and analysis can be performed using commercially available software in conjunction with the ETIS. For those tools specific to multirate systems, including multirate compensator synthesis, documentation for custom software has been provided in Attachment 4.

3. APPLICATION OF THE MULTIRATE DESIGN METHODOLOGY TO THE DESIGN OF A FLUTTER SUPPRESSION SYSTEM FOR THE BACT WING

3.1. INTRODUCTION

To demonstrate some of the advantages of multirate control and the capabilities of our design methodology, we designed several flutter suppression systems for NASA's BACT wing using the methodology in Section 2. A summary of our designs is presented in the following paragraphs. In Section 3.2 we describe the model wing and its open-loop characteristics. In Section 3.3 we discuss our design goals and constraints. In Section 3.4 we discuss our design approach and the details of the design process. In Section 3.5 we present our flutter suppression system design results. Finally, we end the chapter vith some concluding remarks in Section 3.6.

3.2. THE MODEL WING AND ITS OPEN-LOOP DYNAMICS

3.2.1. Model Wing Description

The BACT wing was developed by NASA Langley for the Benchmark Models Program. It consists of a rigid airfoil mounted on a flexible base. The base, called the *Pitch and Plunge Apparatus* (PAPA), provides the two degrees of freedom needed to model classical wing flutter. Our designs used the single control surface (CS) located on the trailing edge of the airfoil and two accelerometers, one near the trailing edge (TE) of the airfoil and one near the leading edge (LE). A diagram of the BACT wing is shown in Fig. 3.1. A detailed description of the BACT wing can be found in [Durham, Keller, Bennett & Wieseman 1991] and [Bennett, Eckstrom, Rivera, Dansberry, Farmer & Durham 1991].

The flutter suppression system was designed using a 16^{th} order linear state model of the BACT wing developed by NASA Langley's Structural Dynamics Division. This model consists of 4 rigid body states corresponding to the pitch and plunge modes, 6 unsteady aerodynamic states, a second order actuator model, a second order Dryden filter, and two first order anti-aliasing filters. A block diagram of the mathematical model is shown in Fig. 3.2 on the following page.

We were provided with 24 different mathematical models of the wing. These models describe the motion of the wing in freon at 24 different operating points. The operating points include dynamic pressures above and below the critical flutter pressure at three different mach numbers. See Table 3.1 on the following page for a summary of the operating points.

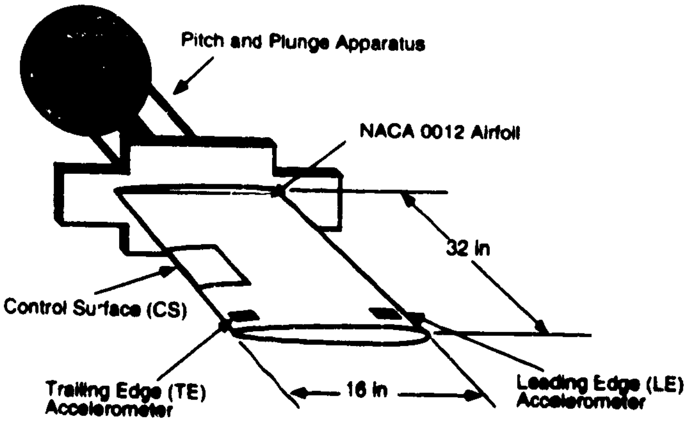

Pitch and Plunge Apparatus
NACA 0012 Airfoil
32 in
Control Surface (CS)
Trailing Edge (TE) Accelerometer
16 in
Leading Edge (LE) Accelerometer

Figure 3.1. BACT wing

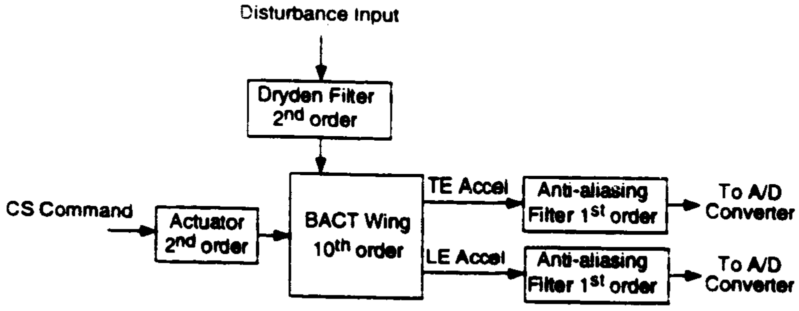


Figure 3.2. Block diagram of BACT wing

Table 3.1. Operating points for BACT wing. All operating points assume Freon medium

	Dynamic Pressure (psf) (Nominally unstable operating points are in gray)							
Mach 0.50	75	100	122	**132**	**150**	**175**	**200**	**225**
Mach 0.70	75	100	125	136	**146**	**175**	**200**	**225**
Mach 0.78	75	100	125	141	**151**	**175**	**200**	**225**

3.2.2. Open-Loop Dynamics

The response of the open-loop BACT wing model at each operating point is characterized by two dominant modes - the pitch and plunge modes. The poles associated with pitch and plunge at mach 0.5 and 75 psf are indicated on Figs 3.3-3.4. As the dynamic pressure increases, one pair of these dominant poles moves towards the right half plane and eventually crosses the imaginary axis at the flutter stability boundary. Figures 3.5-3.6 show the migration of these dominant modes as dynamic pressure increases. The locations of the open-loop poles not shown in the figures remain relatively constant.

The dominant pitch and plunge modes are observable at all operating points with either the TE or the LE accelerometer outputs and are controllable at all operating points using the CS command input. The zeros of the CS command to TE accelerometer and the CS command to LE accelerometer transfer functions are shown in Figs. 3.3-3.4 for an operating point of mach 0.5 and 75 psf. As dynamic pressure increases, the non-minimum phase zeros associated with the TE accelerometer migrate into the left half plane. The minimum phase zeros that are associated with the LE accelerometer and located near the dominant poles migrate into the right half plane. See Figures 3.5-3.6.

At low dynamic pressures the transfer functions from CS command input to both the TE and LE accelerometer outputs are non-minimum phase. Non-minimum phase systems are typically more difficult to control than minimum phase systems. An alternative output is one which measures the difference between the two accelerometers. This new output is essentially pitch acceleration. The CS command to pitch acceleration transfer function is *minimum* phase for all operating points. Figure 3.7 shows the locations of the zeros near the pitch and plunge modes as dynamic pressure increases. It turns out that the BACT wing is fairly easy to control using this new output. The problem is that the pitch acceleration output is artificially created and assumes perfect measurement of TE and LE accelerations. In reality there is some uncertainty in the TE and LE acceleration measurements that must be accounted for in any design. Therefore we did not use the pitch acceleration output directly in our designs. We did, however, use the pitch acceleration output to determine an initial stabilizing compensator for the synthesis algorithm. This is discussed further in Section 3.4.3

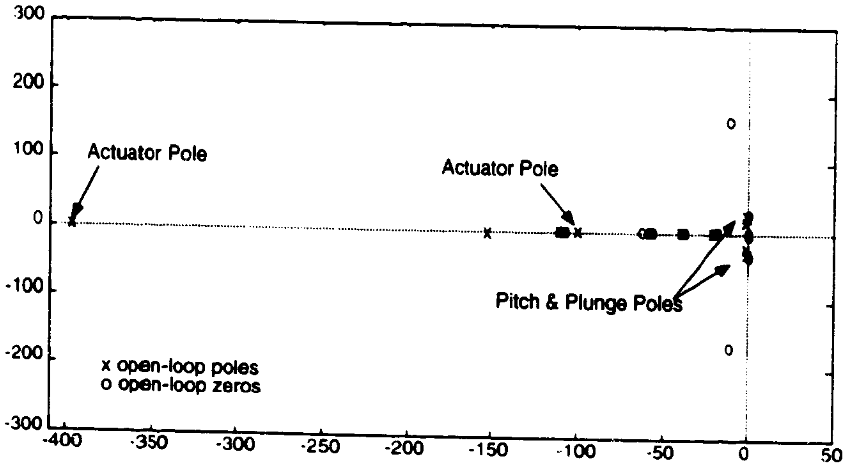


Figure 3.3. Pole/Zero map for open-loop BACT wing at mach 0.50, 75 psf for CS command to TE Accel.

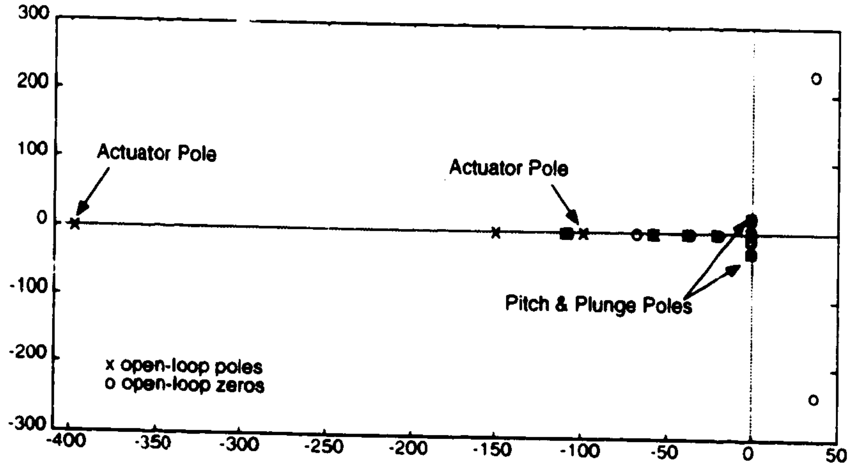


Figure 3.4. Pole/Zero map for open-loop BACT wing at mach 0.50, 75 psf for CS command to LE Accel.

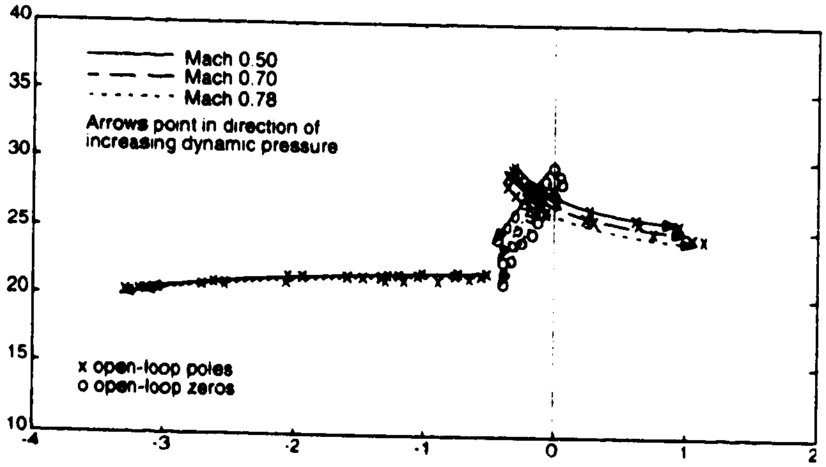


Figure 3.5. Migration of open-loop poles and zeros for CS command to TE Accel.

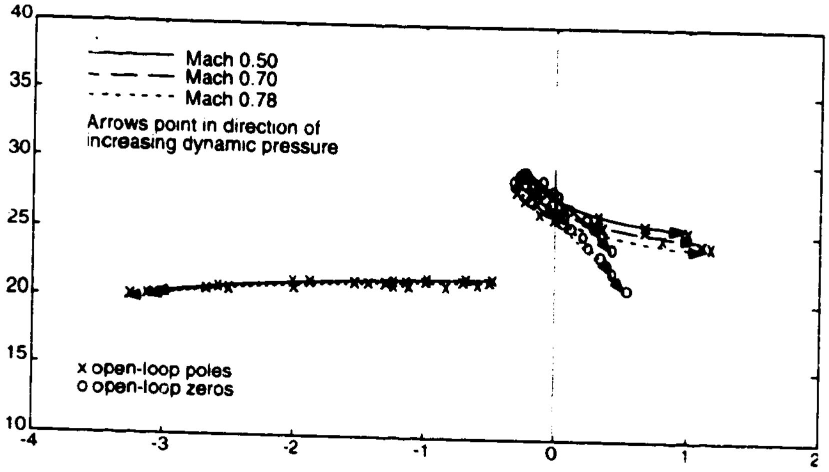


Figure 3.6. Migration of open-loop poles and zeros for CS command to LE Accel.

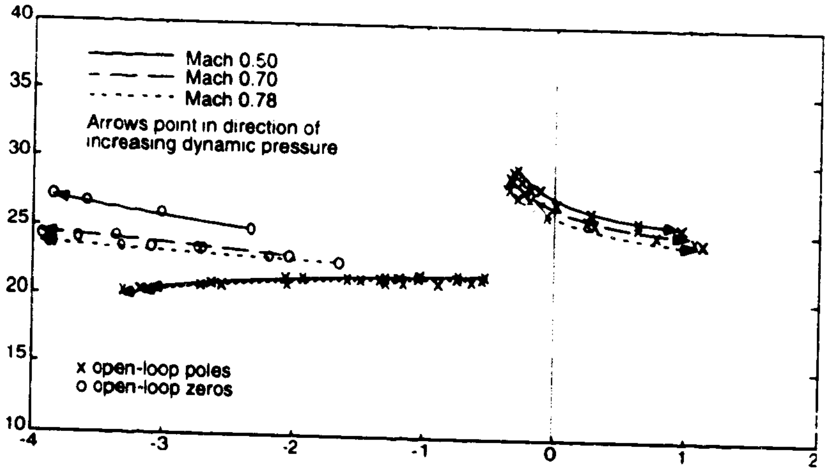


Figure 3.7. Migration of open-loop pole and zeros for CS command to TE-LE Accel.

3.3. DESIGN GOALS AND CONSTRAINTS

The goal of the design was to synthesize a multirate flutter suppression system which stabilizes the BACT wing at all 24 operating points. In addition to stability, NASA Langley specified the following constraints.

Control Activity Constraint: For unity RMS white noise input disturbance (1 in/sec RMS), the steady-state covariance of the CS deflection must not exceed 0.0625 deg^2 (0.25 deg RMS), and the CS deflection rate must not exceed 65 deg^2/sec^2 (8.0 deg/sec RMS).

Sampling Rate Restrictions: The minimum sampling period is 0.005 seconds. For multirate sampling all sampling periods must be multiples of 0.005 sec.

Computational Delay: All compensators must be designed with a minimum 0.005 second computational delay.

Robustness Constraints: The gain and phase margins at the compensator inputs and output must be ±6db and ±45°. At the compensator input, which has two sensors, we use the generalized gain and phase margins based on the singular value. The specified gain and phase margins correspond to a minimum value of 0.75 for the maximum singular value of a multiplicative uncertainty at the compensator inputs (see [Mukhopdhyay & Newsom 1984]).

3.4. FLUTTER SUPPRESSION SYSTEM DESIGN

We used the methodology discussed in Section 2 to design the flutter suppression system. The specific steps for this design were:

1) Select an LQR cost function such that the BACT wing in feedback with the LQ regulator satisfies the criterion specified by NASA
2) Choose an appropriate multirate compensator structure and sampling schedule based on this LQR design
3) Find a set of processor gains so that the compensator stabilizes the BACT wing
4) Synthesize a multirate compensator which minimizes the LQR cost function of step (1) at a few select operating points using the algorithm discussed in Section 2.4
5) Check the performance and robustness of the closed-loop system
6) Iterate on items (1)-(5) as required

We elaborate on the details of each step in the following paragraphs.

3.4.1. Selecting the Cost Function Weights

The multirate synthesis algorithm finds optimum values of the compensator's digital processor gains by minimizing a quadratic cost function with respect to those gains. This optimization can be performed for multiple plant conditions simultaneously. We used the multiple plant capabilities of the algorithm to help ensure that the compensator stabilizes the wing at all 24 operating points. Instead of using all 24 operating points for the optimization we used six representative ones. The six include the operating points at the extremes of mach number and dynamic pressure, and two operating points midway between the extremes. These operating points are listed in Table 3.2 on the following page. For the fault tolerant design discussed in Section 3.4.2.4 we included four additional operating points at mach 0.50. These operating points are grayed in Table 3.2.

For each operating point we selected a unique set of weights for the synthesis algorithm's cost function. The weights were based on a continuous LQR design which weighted the pitch and plunge modes, and the CS command input of the BACT wing. The cost function has the form

$$J = \lim_{t \to \infty} E\left\{ \mathbf{x}(t)^{\mathrm{T}} Q_1 \, \mathbf{x}(t) + u(t)^{\mathrm{T}} Q_2 \, u(t) \right\} \tag{3.1}$$

where $\mathbf{x} = \{x_1 \ x_2 \ x_3 \ x_4\}^{\mathrm{T}}$ and the x_i are the four states associated with the pitch and plunge mode in a modalized version of the BACT wing model. States x_1 and x_2 correspond to the complex conjugate poles which migrate to the left as dynamic pressure increases, see Fig. 3.5. States x_3 and x_4 correspond to the complex conjugate poles which migrate to the right as dynamic pressure increases, see Fig. 3.5. The latter set of poles cause instability in the BACT wing at high dynamic pressures. The variable u is the CS command signal.

For each operating point, the weights, Q_1 and Q_2, were chosen so that the closed-loop damping of the pitch and plunge modes was greater than 0.07, and the RMS control constraints specified by NASA were satisfied. For comparison, the damping in the open-loop BACT wing at the stable dynamic pressure of 75 psf is approximately 0.025. The weights for each operating point were scaled to obtain a unity LQR cost for a 6 inch/sec RMS white noise disturbance input.

Table 3.2. Cost function weights. Grayed operating points used only for fault tolerant design.

Operating Point		State Weight (Q_1)	Control Weight (Q_2)
Mach 0.50	75 psf	diag[$1.2\text{x}10^{-2}$ $1.2\text{x}10^{-2}$ 12 12]	610
Mach 0.50	**132 psf**	**diag[$5.0\text{x}10^{-3}$ $5.0\text{x}10^{-3}$ 3.5 3.5]**	**500**
Mach 0.50	**150 psf**	**diag[$5.0\text{x}10^{-3}$ $5.0\text{x}10^{-3}$ 2.5 2.5]**	**750**
Mach 0.50	**175 psf**	**diag[$4.5\text{x}10^{-3}$ $4.5\text{x}10^{-3}$ 1.4 1.4]**	**900**
Mach 0.50	**200 psf**	**diag[$5.8\text{x}10^{-3}$ $5.8\text{x}10^{-3}$ 0.58 0.58]**	**1754**
Mach 0.50	225 psf	diag[$9.6\text{x}10^{-4}$ $9.6\text{x}10^{-4}$ $9.6\text{x}10^{-2}$ $9.6\text{x}10^{-2}$]	4800
Mach 0.70	125 psf	diag[$1.3\text{x}10^{-2}$ $1.3\text{x}10^{-2}$ 6.4 6.4]	3900
Mach 0.70	175 psf	diag[$1.9\text{x}10^{-3}$ $1.9\text{x}10^{-3}$ 0.56 0.56]	5600
Mach 0.78	75 psf	diag[$8.8\text{x}10^{-2}$ $8.8\text{x}10^{-2}$ 44 44]	8800
Mach 0.78	225 psf	diag[$3.3\text{x}10^{-4}$ $3.3\text{x}10^{-4}$ $1.6\text{x}10^{-2}$ $1.6\text{x}10^{-2}$]	26000

3.4.2. Selecting the Compensator Structure and Sample Rate

Traditionally, the design of a multirate compensator structure begins with a successive loop closures structure and then incorporates cross feed between the loops as necessary. As discussed in Section 2.4.1, multirate successive loop closures is best applied to problems in which the closed-loop system dynamics can be separated into some fast dynamics and some slow dynamics. The BACT wing however does not exhibit those closed-loop characteristics. Closed-loop bode plots, from control input to accelerometer outputs of the BACT wing in feedback with a LQ Regulator, are shown in Fig. 3.8. The LQ Regulator was designed using the cost function weights for the mach 0.50 75 psf operating point specified in Table 3.2. Therefore the bode plot is representative of the closed-loop dynamics we are trying to achieve with the flutter suppression system. Notice that the closed-loop dynamics have only one peak - that associated with the pitch and plunge modes - and do not exhibit the fast and slow dynamics traditionally associated with successive loop closures. Consequently, a traditional multirate successive loop closure structure is not directly applicable to this problem.

Instead of basing our multirate compensator structure on the closed-loop dynamics of the system, we selected compensator structures which used different sampling schedules to reduce either the number of computations or the hardware required to implement the compensator. We designed four compensators: a single-rate (SR); a multirate successive loop closures type (MRSLC); a multirate with multiplexed inputs (MRMI); and a single-rate fault tolerant (SRFT). All of these compensators are second order except the fault tolerant design which is fourth order.

3.4.2.1. Single-Rate (SR)

The single-rate compensator was designed for comparison with the other compensators. A block diagram of this compensator is shown in Fig. 3.9. The sample/update rate for this compensator is 50 Hz. This rate is approximately 10 times the frequency of the dominant pitch and plunge modes. The compensator includes a 0.02 second computational delay, which satisfies NASA's computational delay requirement. This was achieved by constraining the compensator's direct feedthrough term to be zero.

The state space structure of the compensator is

$$\begin{Bmatrix} \xi_1(m,n+1) \\ \xi_2(m,n+1) \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \begin{Bmatrix} \xi_1(m,n) \\ \xi_2(m,n) \end{Bmatrix} + \begin{bmatrix} 0 & b_1 \\ 1 & b_2 \end{bmatrix} \begin{Bmatrix} \text{TE Accel}(m,n) \\ \text{LE Accel}(m,n) \end{Bmatrix} \tag{3.2a}$$

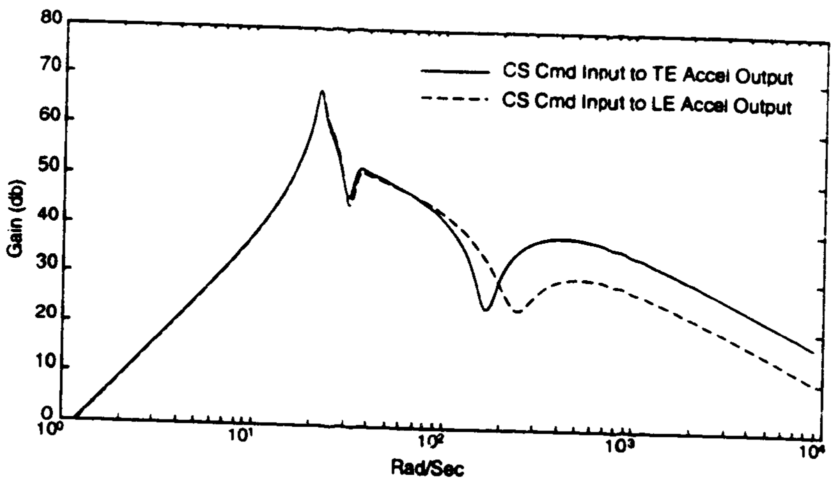


Figure 3.8. Bode plot of closed-loop BACT wing with LQ regulator at mach 0.5 75 psf

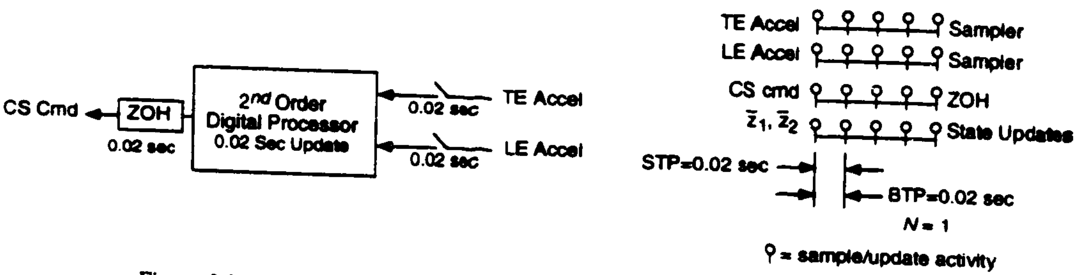


Figure 3.9. Block diagram and corresponding sampling schedule for the SR compensator

$$\text{CS Cmd}(m,n) = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{Bmatrix} \bar{z}_1(m,n) \\ \bar{z}_2(m,n) \end{Bmatrix} \tag{3.2b}$$

where $\bar{z}_1$ and $\bar{z}_2$ are the digital processor states; TE Accel and LE Accel are the acceleration inputs from the A/D converters; and CS Cmd is the command output to the zero-order-hold. a_i, b_i, and c_i are the free gains (matrix elements) which were optimized. The other gains were constrained to the values shown. The structure in Eqn. (3.2) is a minimum realization of the second order compensator. See [Berg, Mason & Yang 1991] for a discussion of minimum realizations. The sampling schedule for Eqn. (3.2) is shown in Fig. 3.9.

3.4.2.2. Multirate Successive Loop Closures (MRSLC)

The MRSLC compensator was designed to reduce the total number of multiplications per unit time performed by the compensator's digital processor. The compensator is comprised of two first order loops. Both loops have two inputs, TE and LE acceleration, and one output, CS command. One of the loops is sampled/updated at 50 Hz, the same as the single-rate design, and the other is sampled/updated four times slower at 12.5 Hz. Just as in the single-rate design, the direct feedthrough terms were constrained to be zero, resulting in a 0.02 second computational delay.

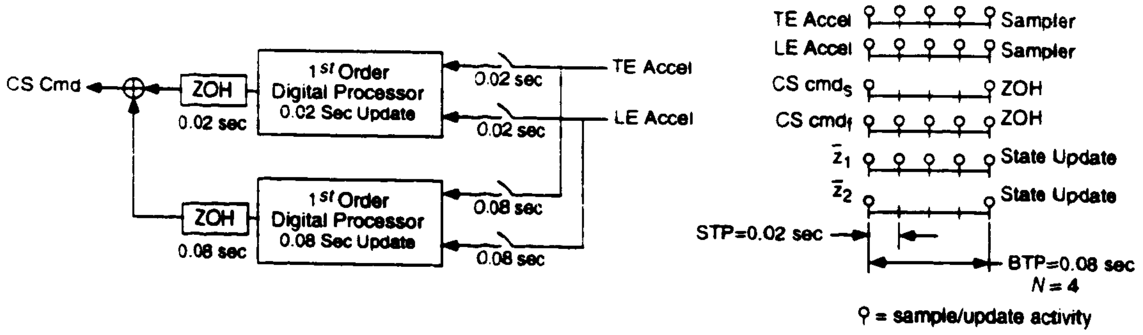


Figure 3.10. Block diagram and corresponding sampling schedule for the MRSLC compensator

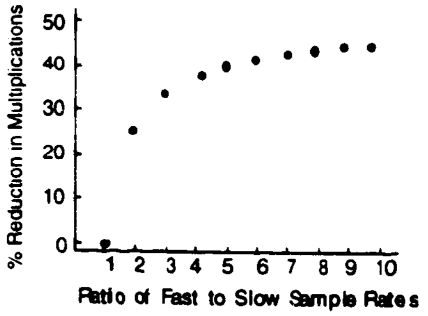


Figure 3.11. Computational savings with the MRSLC design

The net result of this two loop configuration is a compensator structure just like the single-rate design except that the digital processor needs to update one of the digital processor states only every fourth sample/update period. A block diagram of this compensator along with a diagram of its sampling schedule is shown in Fig. 3.10. Note that this diagram only illustrates the structure of the compensator - it is not a schematic of how the compensator would be implemented. When actually implemented, this compensator will use the same number of D/A and A/D converters as the SR compensator, but will require 37% fewer real-time multiplications per unit time.

The choice of sample/update rates for the slow loop was arbitrary within the constraints of the GMCLS. Our goal was simply to reduce the number of multiplications required by the compensator without significantly degrading its performance. The 12.5 Hz sample/update rate was chosen because it is a good compromise between the total number of multiplications saved by utilizing this multirate structure and the ratio of the fast to slow sampling rates. Figure 3.11 shows the percent reduction in the number of multiplication by using the MRSLC design over the SR design. There is a decreasing return in computational savings as the ratio of the fast to slow sampling rate increases. In the limit, the compensator degenerates to a first order compensator with a reduction in multiplications of 50%. Based on Fig. 3.11 we chose a sampling rate ratio of 4.

The state space structure of the compensator which was used for the optimization is

$$\begin{Bmatrix} \bar{z}_1(m,n+1) \\ \bar{z}_2(m,n+1) \end{Bmatrix} = \begin{bmatrix} a_f & 0 \\ 0 & a_s \end{bmatrix} \begin{Bmatrix} \bar{z}_1(m,n) \\ \bar{z}_2(m,n) \end{Bmatrix} + \begin{bmatrix} 1 & b_1 \\ 1 & b_2 \end{bmatrix} \begin{Bmatrix} \text{TE Accel}(m,n) \\ \text{LE Accel}(m,n) \end{Bmatrix} \tag{3.3a}$$

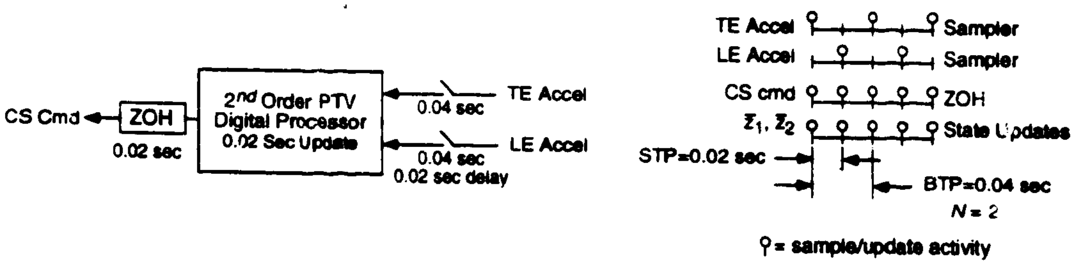


Figure 3.12. Block diagram and corresponding sampling schedule for the MRMI compensator

$$\begin{Bmatrix} \text{CS Cmd}_f(m,n) \\ \text{CS Cmd}_s(m,n) \end{Bmatrix} = \begin{bmatrix} c_f & 0 \\ 0 & c_s \end{bmatrix} \begin{Bmatrix} \bar{z}_1(m,n) \\ \bar{z}_2(m,n) \end{Bmatrix} \tag{3.3b}$$

$$\text{CS Cmd}(m,n) = \text{CS Cmd}_f(m,n) + \text{CS Cmd}_s(m,n) \tag{3.3c}$$

where $\bar{z}_1$ and $\bar{z}_2$ are the digital processor states; TE Accel and LE Accel are the acceleration inputs from the A/D converters; and CS Cmd is the command output to the zero-order-hold. a_i, b_i, and c_i are the free gains which were optimized. The other gains were constrained to the values shown. The structure in Eqn. (3.3) corresponds to the successive loop closures structure of Fig. 3.10. The intermediate outputs CS Cmp$_f$ and CS Cmp$_s$ were added to ensure that Eqn. (3.3) corresponded to Fig. 3.10.

3.4.2.3. *Multirate with Multiplexed Inputs (MRMI)*

The multirate compensator with multiplex inputs was designed to reduce the number of A/D converters required to implement the SR design. In this design, the compensator state and output updates occur at 50 Hz. The outputs of the TE and LE accelerometers are sampled at 25 Hz with a 0.02 second delay between the sampling of the TE accelerometer output and the LE accelerometer output. Thus, the MRMI requires only one A/D converter to sample both accelerometer outputs because it can be multiplexed between the two signals. In addition, the digital processor gains for the MRMI compensator are periodically time-varying. One set of gains is used when the TE accelerometer output is sampled and another set is used when the LE accelerometer output is sampled. Just as in the single-rate design, the direct feedthrough terms were constrained to be zero, resulting in a 0.02 second computational delay. This compensator requires the same number of multiplications per unit time as the SR design but it uses only one D/A converter. Figure 3.12 shows a block diagram of the MRMI compensator.

The state space structure of the MRMI compensator is

$$\begin{Bmatrix} \bar{z}_1(m,n+1) \\ \bar{z}_2(m,n+1) \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1(n) & a_2(n) \end{bmatrix} \begin{Bmatrix} \bar{z}_1(m,n) \\ \bar{z}_2(m,n) \end{Bmatrix} + \begin{bmatrix} 0 & b_1(n) \\ 1 & b_2(n) \end{bmatrix} \begin{Bmatrix} \text{TE Accel}(m,n) \\ \text{LE Accel}(m,n) \end{Bmatrix} \tag{3.4a}$$

$$\text{CS Cmd}(m,n) = \begin{bmatrix} c_1(n) & c_2(n) \end{bmatrix} \begin{Bmatrix} \bar{z}_1(m,n) \\ \bar{z}_2(m,n) \end{Bmatrix} \tag{3.4b}$$

where $\bar{z}_1$ and $\bar{z}_2$ are the digital processor states; TE Accel and LE Accel are the acceleration inputs from the A/D converters; and CS Cmd is the command output to the zero-order-hold. $a_i(n)$, $b_i(n)$, and $c_i(n)$ are the free gains which were optimized. These gains are functions of n because they are periodically time-varying, e.g. $a_i(n) = a_i(n+2)$ The other gains were constrained to the values shown. The sampling schedule for Eqn. (3.4) is shown in Fig. 3.12.

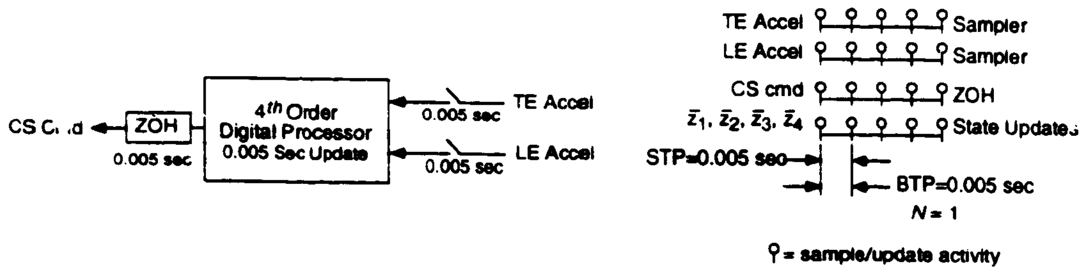


Figure 3.13. Block diagram and corresponding sampling schedule for the SRFT compensator

3.4.2.4. *Single-Rate Fault Tolerant (SRFT)*

The single-rate fault tolerant compensator was designed to highlight the multiple plant capability of our synthesis algorithm. This compensator is fourth order with a sample/update rate of 200 Hz and a 0.005 second computational delay. A block diagram of the compensator and its corresponding sampling schedule are shown in Fig. 3.13. The state space representations of the SRFT compensator is similar to the 2^{nd} order single-rate compensator with the exception that the digital processor is fourth order.

The SRFT compensator is fault tolerant in the sense that it stabilizes all the plant conditions even with one of the accelerometers disconnected. To achieve fault tolerance for all 24 plant conditions, we optimized the compensator for 22 simultaneous plant conditions - as opposed to just six for the preceding designs. These include the six operating points used in the previous designs evaluated at three cases each: 1) both TE and LE sensors active; 2) only the TE sensor active; and 3) only the LE sensor active. In addition to those 18, we added four more operating points at mach 0.50 evaluated for the case where only the LE sensor is active. These operating points are grayed in Table 3.2.

3.4.3. Designing a Stabilizing Compensator

We used the synthesis algorithm presented in Section 2.4 to optimize the gains of the four compensators discussed in Section 3.4.2. The algorithm requires an initial guess for the compensator's digital processor gains for which the closed-loop system, the BACT wing and compensator, is stable. The difficulty in finding these gains is that the closed-loop system must be stable at all operating points used in the optimization.

To get a stabilizing guess for the wing at all operating points we used a boot-strapping technique. First we found values of the processor gains which stabilized the BACT wing at *one* operating point. Then we optimized the gains for the wing at that *one* operating point using large values for the plant disturbance noise and sensor noise intensities. The large value of noise intensities introduced uncertainty into the plant. Consequently, the resulting compensator was more robust than a compensator optimized for a plant with no noise. This new set of processor gains always stabilized the wing at the original operating point plus at least one other operating point. We then used the new processor gains as the initial guess to the problem with the wing at two (or more) operating points. The procedure was continued until the compensator stabilized the plant at all the operating points and the problem could be solved using realistic noise intensities.

Before beginning the bootstrapping procedure we needed to find a set of processor gains which stabilized the closed-loop system for at least one operating point. This was straightforward for the SR and MRSLC compensators. We designed a first order single-rate compensator with pitch acceleration input, CS command output and a sampling rate of 50 Hz. Recall from Section 3.2.2 that pitch acceleration is essentially the difference in the TE and LE accelerations. The pole location and gain value of this compensator were found using root locus. The initial stabilizing guess for the SR design consisted of this first order compensator in parallel with an arbitrary first order compensator that had an input/output gain of zero. For the MRSLC system, we used the first order compensator as an initial guess for the fast loop of the successive loop closures structure, and an arbitrary first order compensator, with an input/output gain of zero, for the slow loop.

An initial guess for the MRMI processor gains was more difficult to find than for the SR and MRSLC compensators. Due to its complex sampling schedule we could not design an initial guess by traditional methods. Instead, we designed a compensator with the multiplexed structure but with very small gains. Then we used the bootstrapping technique, beginning with the BACT wing operating at a low dynamic pressure where it is open-loop stable. Since the compensator gains were very small, they did not destabilize the wing and could be used as an initial guess. The bootstrapping process for this compensator took several iterations, verses one or two for the other compensators, because we began with such a poor initial guess.

To obtain an initial guess for the SRFT processor gains we began by designing two 2^{nd} order compensators. One stabilized the plant when the LE sensor was disconnected, the other stabilized the plant when the TE was disconnected. We then combined these two compensators into a single 4^{th} order design and adjusted their gains until the new fourth order compensator stabilized the plant when both sensors were active or when only one or the other was active. Finally this design was used in the bootstrapping procedure discussed earlier to obtain a single fourth order compensator which stabilized the wing at all operating points.

3.4.4. Optimizing the Digital Processor Gains

We optimized the digital processor gains of the three compensators with the algorithm discussed in Section 2.4. The optimization used the following parameters:

Plant Conditions: Six simultaneous operating points for the second order designs; 22 simultaneous operating points for the fourth order design. See Table 3.2 and Section 3.4.2.4

Cost Function Weights: The second order designs used the cost function weights listed in Table 3.2. The fourth order design used the weights in Table 3.2 for cases where both the TE and LE sensors were active, and one-tenth those values for cases where either sensor was inactive

Process Noise PSD value: 36 in^2/sec^2 - this is the intensity of the white noise input to the Dryden filter and was specified by NASA

Sensor Noise PSD value: 0 rad^2/sec^4 for initial designs, 240 rad^2/sec^4 for final designs. This is discrete sensor noise for the TE and LE acceleration measurements

Initial Stabilizing Gains: Obtained using root locus and bootstrapping, see Section 3.4.3

Compensator Structure: See equations (3.2)-(3.4)

Sampling Schedule: See Figures 3.9, 3.10, 3.12 and 3.13.

Gain Constraints: In all designs the direct feed through terms were constrained to be zero. Additional gain constraints for each compensator are specified in Section 3.4.2.

The M-Files which define the above input parameters for the synthesis software presented in Attachment 4 are documented in Appendix B

3.4.5. Design Iteration Based on Performance and Robustness Analysis

After synthesizing the multirate compensators we evaluated their performance and robustness using the methods discussed in Section 2.5. One of the robustness measures was the maximum singular value of the minimum destabilizing multiplicative uncertainty at the compensator inputs (a structured singular value). When we synthesized the compensators using a sensor noise covariance intensity of zero, the size of the destabilizing gain was unacceptably small - less than 0.20 for the BACT wing at some operating points. NASA had specified a value of 0.75. To improve the robustness at the compensator input we increased the sensor noise intensity to 240 rad^2/sec^4 and re-optimized the processor gains. This procedure was motivated by the Loop Transfer Recover technique for LQG systems described in [Doyle & Stein 1981]. The results of increasing the sensor noise are discussed in the following Section.

3.5. DESIGN RESULTS

We designed four compensators using the approach discussed in the previous sections. For review, the four are the:

1) Single-Rate 2nd Oraer (SR)
2) Multirate 2nd Order Successive Loop Closures (MRSLC)
3) Multirate 2nd Order Multiplexed Input (MRMI)
4) Single-Rate Fault Tolerant (SRFT)

The structure of each of these compensators was discussed in Section 3.4.2. Optimum values for the digital processor gains are given in Appendix A.

We looked at five performance and robustness measures:

1) Cost function value
2) Gust pulse response
3) Maximum RMS gain from disturbance to the control surface deflection and deflection rate
4) Gain and phase margins at the compensator output
5) The maximum singular value of the minimum destabilizing multiplicative uncertainty at the compensator input

Results are presented for three operating points, mach 0.50 132 psf, mach 0.70 146 psf, and mach 0.78 151 psf. Each of these operating points is 5 psf above the critical flutter dynamic pressure for the corresponding mach number, and so the BACT wing is nominally unstable at each of these operating points. It is important to note that none of these operating points were used for the compensator optimization. Therefore the compensators were not tuned to these particular operating points. In general, the performance and robustness of the compensators at these three operating points is indicative of their performance at the remaining 21 operating points.

3.5.1. Cost Function Value

One measure of the overall steady-state performance of a compensator is the value of the cost function in Eqn (3.1) at the optimum value of the digital processor gains. (A value for the cost function is returned by our synthesis algorithm at the completion of the optimization.) For our 2nd order designs, a "perfect" compensator would have a cost function value of 6, assuming no sensor noise. The "perfect" fault tolerant design would have a cost of 7.6 since it optimizes a different cost function. By "perfect" compensators we mean continuous LQR designs with gain scheduling, i.e., they use a different set of feedback gains at every operating point. We expect the costs associated with our compensators to be higher since they used discrete sampling, did not use gain scheduling, and had fictitious sensor noise.

It is more realistic to compare the cost of our compensators to that of a discrete LQG design with fictitious sensor noise and gain scheduling. This comparison eliminates some of the differences due to sampling and fictitious sensor noise. The cost associated with the discrete LQG compensator is the lowest cost we can expect for a given sampling rate and sensor noise level. Table 3.3 summarizes the values of the cost function for the discrete LQG and for our four designs. The costs associated with our second order compensators are almost twice that of the discrete LQG design. This is not surprising since the discrete LQG is significantly more complex - it is a 16^{th} order compensator with gain scheduling.

3.5.2. The Gust Pulse Response

The gust pulse response provides an indication of the transient response of the closed-loop system due to a disturbance input. The gust pulse response was found by simulating the response of the BACT wing in feedback with the flutter suppression system to a disturbance input pulse with an amplitude of 10 in/sec and a duration of 0.004 seconds. This simulation was performed using the M-file **mrsim** described in Attachment 4.

Figures 3.14-3.16 show the response of the BACT wing at mach 0.70 and 146 psf to the specified disturbance gust pulse. Also shown is the response of the wing with a continuous LQ regulator. The cost function weights for this LQ regulator design satisfy the same design criterion as was used to optimize the compensator's gains. (See Section 3.4.1.) We provided response plots for only one operating point. The gust pulse responses at other operating points are similar to those provided in Figs. 3.14-3.16.

For comparison we also provided a gust pulse response plot for the 2^{nd} order compensators synthesized without fictitious sensor noise. Recall that fictitious noise was added to the sensors in order to improve the robustness at the compensator input. Figure 3.17 shows the pitch response of the BACT wing at mach 0.70 and 146 psf due to a gust pulse disturbance. The primary effect of adding sensor noise is to decrease the damping of the pitch and plunge modes. The reduction in damping is more prevalent in the pitch response than in the plunge response.

The gust pulse response plots are shown below.

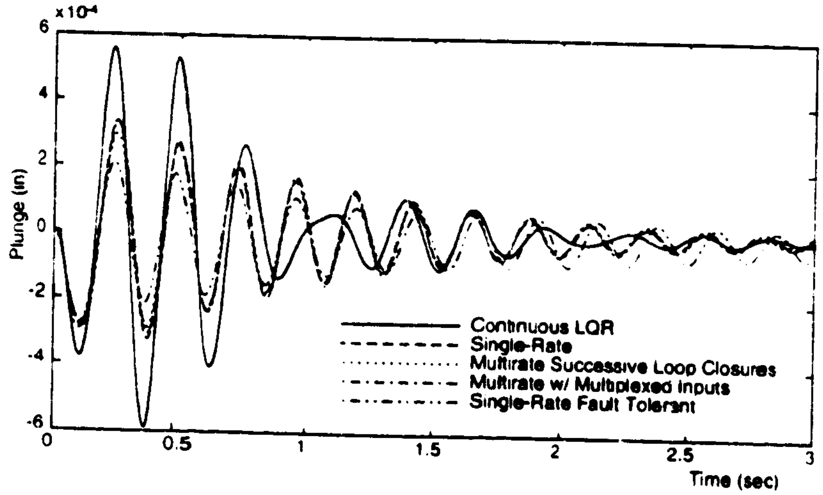


Figure 3.14. Plunge gust pulse response at mach 0.70 146 psf

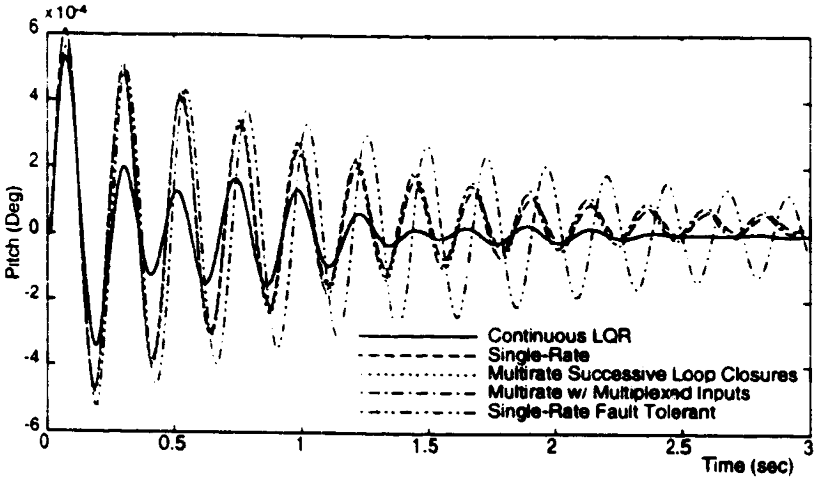


Figure 3.15. Pitch gust pulse response at mach 0.70 146 psf

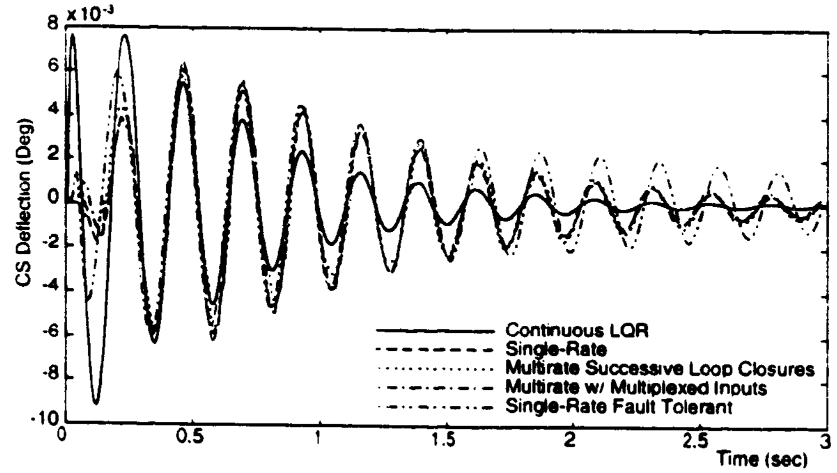


Figure 3.16. Control surface deflection gust pulse response at mach 0.70 146 psf

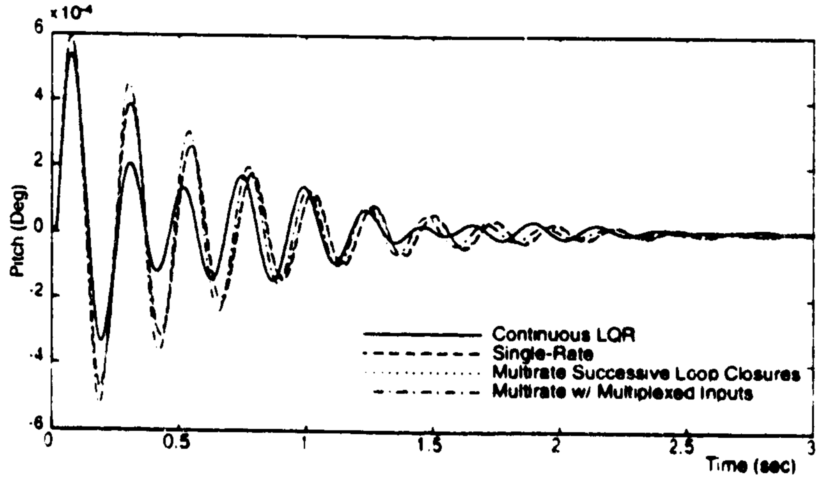


Figure 3.17. Pitch gust pulse responses at Mach 0.70 146 psf for compensator designed without fictitious sensor noise

Table 3.3 Design Results

	Mach 0.50, 132 psf				Mach 0.70, 146 psf				Mach 0.78, 151 psf			
	SR	MRSLC	MRMI	SRFT	SR	MRSLC	MRMI	SRFT	SR	MRSLC	MRMI	SRFT
Cost Function Value[1]	13.9[2]	14.2[2]	15.6[2]	26.0[3]								
Max RMS Gain Distr. to CS Deflect. (deg sec/in)	0.22	0.22	0.25	0.22	0.19	0.19	0.19	0.13	0.11	0.11	0.11	0.11
Distr. to CS Def. Rate (deg/in)	6.5	6.6	6.9	6.2	2.4	2.8	2.6	3.4	1.5	1.6	1.5	2.8
Gain Margin	12 db	14 db	12 db	8 db	10 db	12 db	10 db	6 db	9 db	11 db	9 db	5 db
Phase Margin	41°	45°	38°	21°	45°	48°	40°	18°	43°	48°	40°	15°
$\bar{\sigma}\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$	0.41	0.49	0.45	0.35	0.38	0.48	0.44	0.28	0.35	0.49	0.45	0.25
$\bar{\sigma}\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$ w/o sensor noise	0.25	0.25	0.35	not calculated	0.26	0.23	0.32	not calculated	0.25	0.21	0.31	not calculated

[1] Cost Function Value is not a function of the operating point

[2] The corresponding discrete LQG cost is 7.7

[3] The corresponding discrete LQG cost is 8.6

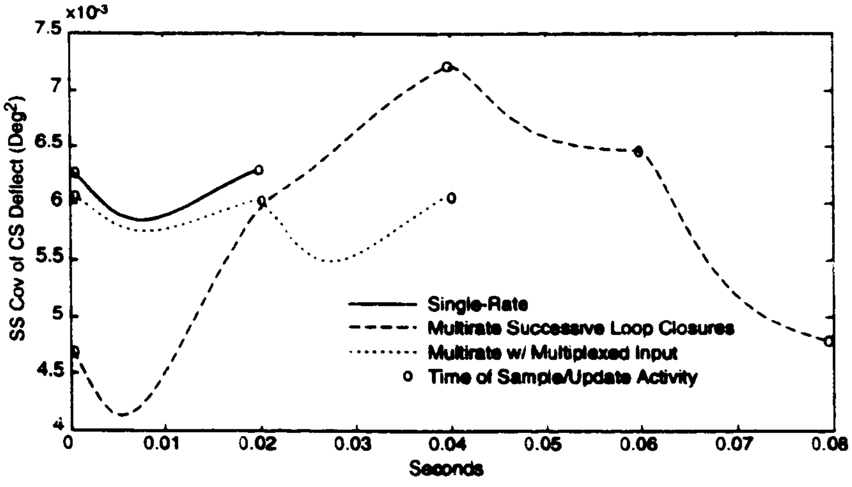


Figure 3.18. Steady-state covariance propagation at mach 0.70 146 psf with 1 in/sec RMS white noise disturbance

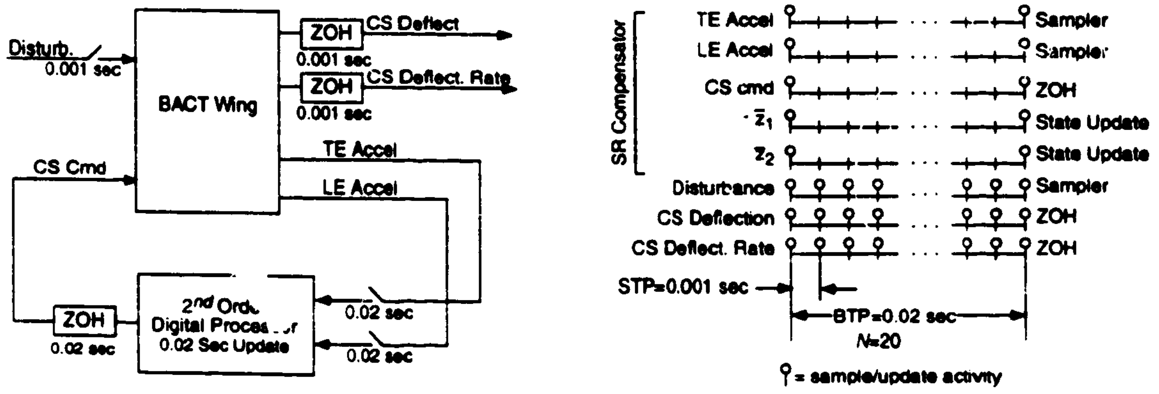


Figure 3.19. Block diagram of discrete system for calculating RMS gain and corresponding sampling schedule

3.5.3. RMS Gain for Control Surface Deflection and Deflection Rate

One of NASA's specifications was a limit on the steady-state covariance of the control surface deflection and deflection rate for a 1 in/sec RMS white noise disturbance. Our closed-loop system consists of a continuous plant and a discrete compensator. Therefore these steady-state covariances are periodically time-varying. In Fig. 3.18 we show the steady-state covariance propagation for the BACT wing in feedback with the three compensators at an operating point of mach 0.70 and 146 psf for a unity RMS white noise disturbance.

We calculated the values of the steady-state covariance at the sample/update times using the method described in Section 2.5.4. Between the sample/update times of the compensator, the covariances were propagated using the dynamics of the open-loop continuous BACT wing. The steady-state covariances are only shown for one BTP of the compensator - they repeat themselves during every BTP of the compensator.

One meaningful interpretation of NASA's specification would be to look at the peak steady-state covariance value taken from this covariance plot. This value, though, is an upper limit on the closed-loop gain for a white noise disturbance and is not an accurate indicator of the control activity level. A better measure of control activity would be the maximum RMS gain calculated using Eqn. 2.25. This is an exact measure of the maximum RMS gain for *any* non-decaying input signal.

In order to apply Eqn. (2.25), which is for a discrete system, to our mixed continuous/discrete system we created a new discrete multirate system in which the continuous inputs and outputs of interest are sampled very fast (see Section 2.5.3). We chose a sampling rate for the CS deflection and deflection rate of 1000 Hz. This is more than twenty times the control surface actuator rolloff frequency. A block diagram of this new discrete-time system, with the single-rate compensator of Eqn. (3.2), is shown in Fig. 3.19 along with its sampling schedule. This new system is now *multirate* even though the compensator is *single-rate*. The ETIS for this system has a sample/update rate of 1000 Hz and an N of 20.

We used this new ETIS system to calculate the maximum RMS gain of the original system between the disturbance and the CS deflection and between the disturbance and the CS deflection rate. The maximum RMS gains for the BACT wing at three operating points are summarized in Table 3.3. See also the related work of [Sivashankar & Khargonekar 1991].

3.5.4. Gain and Phase Margins at the Compensator Output

Gain and phase margins were calculated at the compensator output using the ETIS and a multiloop Nyquist diagram. The ETIS of the plant and compensator were computed independently and then combined in series to form an ETIS loop transfer function. Gain and phase margins were subsequently measured directly off the multiloop Nyquist plot of this function. These are traditional gain and phase margins, and assume that the gain and phase do not vary simultaneously. The details of this technique are given in Section 2.5.1, [Mason 1992], and [Mason & Berg 1992] (Attachment 2).

The gain and phase margins for the BACT wing at three operating points are presented in Table 3.3. These values are typical of the margins at all 24 operating points, although the margins tend to be better at lower dynamic pressures and slightly worse at higher dynamic pressures. A representative Nyquist diagram is shown in Fig. 3.20. This particular Nyquist plot has two encirclements of the −1 point because the open-loop plant has two unstable poles.

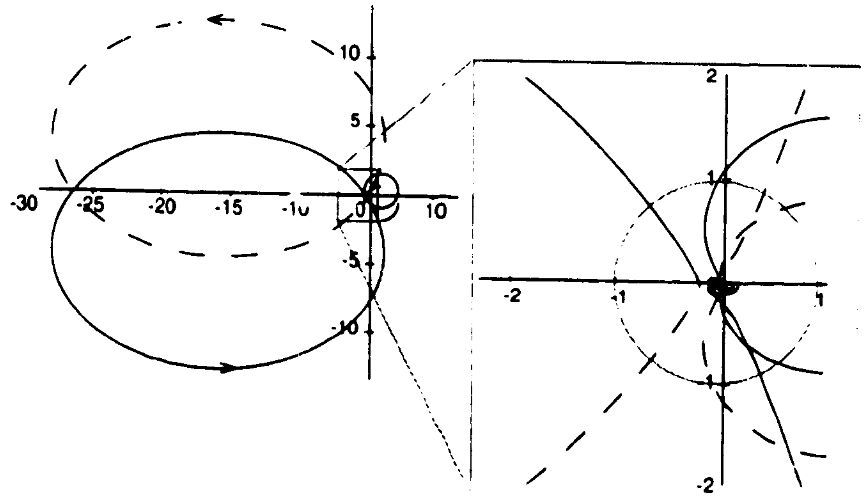


Figure 3.20. Multiloop Nyquist for BACT wing at mach 0.70 146 psf with MRMI compensator

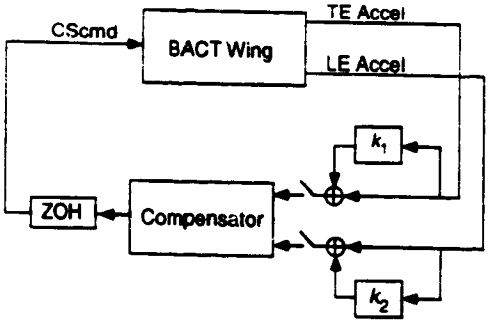
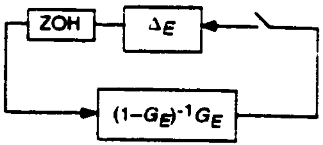


Figure 3.21. Uncertainty Model



$$\Delta_E = \text{block diag}\left\{\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}, \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}, \cdots, \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}\right\} \text{ with } N \text{blocks}$$

Figure 3.22 ETIS uncertainty output feedback model

3.5.5. Robustness at the Compensator Input

The uncertainty at the compensator input was assumed to be a multiplicative perturbation of the form shown in Fig. 3.21, where k_1 and k_2 are complex gains. We transformed this system into the output feedback form traditionally used in robustness analysis using simple block diagram algebra. However, when the compensator is multirate we must use the ETIS of the plant, compensator and uncertainty. A block diagram of this closed-loop ETIS for the multirate flutter suppression system is shown in Fig. 3.22. G_E is the loop transfer function consisting of the compensator and plant ETIS transfer functions connected in series.

Now, given the system in the form shown in Fig. 3.22, we can calculate an exact value for the size of the smallest destabilizing perturbation [Doyle 1982] . First rewrite Δ_E in Fig. 3.22 as

$$\Delta_E = I_1 k_1 + I_2 k_2 \tag{3.5}$$

where I_1 = diag{1 0 1 0 ... 1 0} with $2N$ diagonal elements, and where I_2 has a similar form. Then it can be shown that

$$\bar{\sigma}(\Delta_{\min}) = \left(\sup_{\phi} \max_{\theta} \rho\left\{\left[I_1 + I_2 e^{j\theta}\right] H_E(e^{j\phi})\right\}\right)^{-1} \text{ for } 0 \le \phi \le \pi \text{ and } 0 \le \theta \le 2\pi; \tag{3.6}$$

where $\bar{\sigma}(\Delta_{\min})$ represents the maximum magnitude of the smallest destabilizing k_1 or k_2; ρ is the spectral radius; and $H_E(z^N) = (I - G_E(z^N))^{-1} G_E(z^N)$.

We are guaranteed that the system in Fig. 3.21 will remain stable as long as

$$\bar{\sigma}\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} < \bar{\sigma}(\Delta_{\min}) \tag{3.7}$$

We are also guaranteed that when Eqn. (3.7) is violated, there exist values of k_1 and k_2 that destabilize the system in Fig. 3.21.

Equation (3.6) is straightforward to solve with a two dimensional search in ϕ and θ. The results are given in Table 3.3. For comparison, the corresponding results for the design without the fictitious sensor noise are also given in Table 3.3 Notice that the addition of the fictitious noise increases the maximum singular value of the smallest destabilizing uncertainty by as much as 60%.

Even with the fictitious sensor noise, the robustness at the compensator inputs does not meet NASA's specification for a maximum singular value of 0.75. We could have improved the robustness at the compensator output further by increasing the fictitious sensor noise level, but we chose not to do so because this simultaneously reduces the gain and phase margins at the compensator output.

3.6. CONCLUSIONS

The performance and robustness of the three 2^{nd} order compensators are nearly identical. All three stabilize the BACT wing at all 24 plant conditions and, with the exception of the robustness at the compensator input, satisfy all of NASA's specifications. From this perspective there is little reason to use the multirate designs over the single-rate design.

The real advantage of the multirate designs is that they allow the engineer to trade design simplicity for reductions in real-time computations or a reduction in hardware. The successive loop closures design trades a reduction in the number of computations for a more complex digital processor program. The multiplexed design trades one A/D converter for multiplexing hardware and a more complex digital processor program. Depending on the costs of the hardware, such trades might be very advantageous.

The 4^{th} order fault tolerant design, on the other hand, does not satisfy NASA robustness specifications. The compensator does, however, meet the robustness specifications to which it was designed. It stabilizes the BACT wing at all 24 operating point even if one of the accelerometers fails. This type of robustness - to a very specific perturbation - would be difficult to achieve using more common robustness improvement techniques such as Loop Transfer Recovery, but was straightforward to achieve using the multiple plant condition capability of our synthesis algorithm.

4. CONCLUSIONS AND RECOMMENDATIONS

4.1. CONCLUSIONS

The principle advantage of multirate control is that it gives the designer freedom to choose a sampling schedule which best utilizes the available hardware and software. In the flutter suppression system design, for example, we developed multirate controllers that provide performance comparable to a single-rate design, yet require either fewer real time multiplications per unit time to implement or require fewer A/D converters.

The disadvantage of multirate control is that this additional flexibility substantially increases the complexity of design and analysis over the single-rate case. Undoubtedly, the lack of good design and analysis tools has discouraged many from applying multirate control even when the situation may be ideal for a multirate design.

In this report we addressed the difficulty of multirate design and analysis by presenting a multirate design methodology. The methodology specifies a design approach and provides specific tools necessary to apply the approach to a practical problem. The tools are for modeling a multirate system, for synthesizing a multirate compensator which is robust to plant perturbations, and for analyzing the performance and robustness of a multirate system. The resulting methodology is powerful and straightforward to apply.

To demonstrate the methodology we applied it to design several multirate compensators for NASA's BACT wing. Those compensators satisfy the specified design specifications and illustrate some of the benefits of multirate control.

4.2. RECOMMENDATIONS FOR FUTURE RESEARCH

1) Our synthesis algorithm currently requires a stabilizing initial guess for the digital processor gains. Obtaining a stabilizing initial guess for those gains can be difficult, especially when the multiple plant conditions capability of the algorithm is used, because the initial guess must stabilize all plant conditions simultaneously. Eliminating this requirement would substantially improve the algorithm's versatility.

2) The singular value analysis of multirate systems leads directly to a structured singular value problem with repeated blocks. Calculating an exact solution to this problem is difficult for all but the simple tow parameter case. This is an area which needs further research.

REFERENCES

1. Amit, N., and Powell J.D., "Optimal Control of Multirate Systems," *Proc. AIAA Guid. Contr. Conf.*, Albuquerque, NM, 1981.

2. Apostolakis, I.S. and Jordan, D., "Multirate system Performance Evaluation Using Singular Value Analysis," *1990 ACC*, pp.1502-1507.

3. Bennett, R.M., Eckstrom, C.V., Rivera, J.A.Jr., Dansberry, B.E., Farmer, M.G., and Durham, M.H., "The Benchmark Aeroelastic Models Program - Description and Highlights of Initial Results," NASA Tech. Memo. 104180, Dec. 1991

4. Berg, M.C., "Design of Multirate Digital Control Systems," Ph.D. Thesis, Stanford Univ., Stanford, CA, 1986.

5. Berg, M.C., Amit, N. and Powell, J.D., "Multirate Digital Control System Design," *IEEE Trans. Auto. Contr.*, Vol AC-33, Dec 1988, pp.1139-1150.

6. Berg, M.C., and Yang, G.S., "A New Algorithm for Multirate Digital Control Law Synthesis," *Proc. IEEE Conf. Decision Contr.*, Dec 1988, Austin, TX, pp. 1685-1690.

7. Berg, M.C. and Mason, G.S., "Multirate Sampled-Data Yaw-Damper and Modal Suppression System Design," Final Report for NASA Langley Research Grant NAG-1-1055, Feb. 1991.

8. Berg, M.C., Mason, G.S. and Yang, G.S., "A New Multirate Sampled-Data Control Law Structure and Synthesis Algorithm," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 15, No. 5, 1992, pp. 1183-1191.

9. Boyd, S., and Doyle, J., "Comparison of peak and RMS gains for discrete-time systems," *Systems & Control Letters*, vol 9, 1987, pp. 1-6.

10. Boykin, W.H. and Frazier, B.D., "Analysis of Multiloop, Multirate Sampled-Data Systems," *AIAA Jour. Guid. Control & Dyn.*, Vol. 13, May 1975.

11. Dansberry, B.E., "Dynamic characteristics of a Benchmark Models Program Supercritical Wing," AIAA Paper No. 92-2368, April, 1992.

12. Dahleh, M.A., Voulgaris, P.G., and Valavani, L.S., "Optimal and Robust Controllers for Periodic and Multirate Systems," *IEEE Trans. Auto. Control*, Vol. 37, No 1., January 1992, pp. 90-99.

13. Doyle, J., "Analysis of Feedback systems with Structured Uncertainties," *IEE Proc.*, Vol 129, Pt. D, No. 6, Nov. 1982, pp. 243-250.

14. Doyle, J.C., and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," *IEEE Trans. Auto. Control*, Vol AC-26, Mo 1, Feb 1981, pp 4-16.

15. Durham, M.H., Keller, D.F., Bennett, R.M., Wieseman, C.D., "A Status Report on a Model for Benchmark Active Controls Testing," AIAA Paper No. 91-1011, April 1991.

16. Franklin, G.F., Powell, J. David, and Workman, M.L., *Digital Control of Dynamic Systems*, Addison-Wesley Pub. Co. 1990.

17. Glasson, D.P., "A New Technique for Multirate Digital Control Design and Sample Rate Selection," *AIAA Jour. Guid. Control & Dyn.*, Vol. 5, July-Aug. 1982.

18. Glasson, D.P., "Development and Application of Multirate Digital Control," *IEEE Control Systems Magazine*, Vol. 3, No. 4, 1983, pp. 2-8.

19. Kalman, R.E. and Bertram, J., "A Unified Approach to the Theory of Sampling Systems," *Journal of the Franklin Institute*, Vol 267, May 1957.

20. Kono, M., "Eigenvalue assignment in Linear Periodic Discrete-Time Systems," *Int. J. Control*, Vol. [illegible], No. 1, 1980, pp. 149-158.

21. Kranc, G.M. "Compensation of an Error Sampled System by a Multirate Controller," *Part II, AIEE Transactions*, Vol. 76, July 1957.

22. Kranc, G.M., "Input-Output Analysis of Multirate Feedback Systems," *IRE Trans. Auto. Control*, Vol. AC-3, Nov. 1957, pp. 21-28.

23. Kwakernaak, H. and Sivan, R., *Linear Optimal Control Systems*. New York, Wiley, 1972

24. Ly, U.L., "A Design Algorithm for Robust Low-Order Controllers." Ph.D. Thesis, Stanford Univ., Stanford, CA, 1982.

25. MacFarlane, A.G.J., "Return-Difference and Return-Ratio Matrices and Their Use in Analysis and Design of Multivariable Feedback Control Systems," *IEE Proc.*, Vol. 117, No. 10, Oct. 1970, pp.2037-2049.

26. Maciejowski, J.M., *Multivariable Feedback Design*, Addison-Wesley Publishing Company, Reading Mass. 1990.

27. Mason, G.S., "Multirate Compensator Synthesis and Analysis", Ph.D. Dissertation, Univ. Washington. 1993

28. Mason, G.S. and Berg, M.C., "Reduced Order Multirate Compensator Synthesis," *AIAA Jour. Guid. Contr. and Dynamics*, Vol 15, No. 3 May-June 1992, pp. 700-706.

29. Mason, G.S. and Berg, M.C., "Robustness Analysis of a Multirate Flutter Suppression System," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 16, No. 5, 1993, pp. 992-926.

30. Mason, G.S. and Berg, M.C., "Multirate Flutter Suppression System for a Model Wing," accepted for publication in the *AIAA Journal of Guidance, Control and Dynamics*.

31. Meyer, R.A., and Burrus, C.S., "A Unified Analysis of Multirate and Periodically Time-Varying Digital Filters," *IEEE Trans. Circuits in Systems*, Vol. CAS-22, No. 3, March 1975, pp. 162-168.

32. Mukhopadhyay, V., "Digital Robust Control Law Synthesis Using Constrained Optimization," *AIAA Jour. Guid., Contr. and Dynamics*, Vol. 12, March-April 1989, pp. 175-181.

33. Mukhopadhyay, V., and Newsom, J.R., "A Multiloop System Stability Margin Study Using Matrix Singular Values," *AIAA Jour. Guid., Contr. and Dynamics*, Vol. 7, Sept-Oct 1984, pp. 582-587.

34. Mukhopadhyay, V., Newsom, J.R., and Abel, I., "A Method for Obtaining Reduced-Order Control Laws for High-Order Systems Using Optimization," NASA Tech. Paper 1876, August 1981.

35. Narigon, M.L., "Optimal Sample Policies for Multirate Digital Control," Ph.D. Thesis, Stanford Univ., Stanford, CA, 1991.

36. Packard, A., "What's New with MU: Structured Uncertainty in Multivariable Control," Ph.D. Dissertation, Univ. Cal. Berkley, 1988.

37. Safonov, M. and Doyle, J.C., "Minimizing Conservativeness of Robustness Singular Value.," *Multivariable Control: New Concepts and Tools*, 1984, pp. 197-207.

38. Safonov, M.G., "Stability Margins of Diagonally Perturbed Multivariable Feedback Systems," *IEE Proc.*, Vol 129, Pt. D, No. 6, Nov. 1982, pp. 251-256.

39. Sivashankar, N., and Khargonekar, P.P., "Induced Norms for Sampled-Data Systems," submitted to *Automatica* July 1991.

40. Thompson, P.M., "Gain and Phase Margins of Multirate Sampled-Data Feedback Systems," *Inter. Jour. Control*, Vol. 4, No. 3, 1986, pp. 833-846.

41. Yang, G.S., "A Generalized Synthesis Method for Multirate Feedback Control Systems," Ph.D. Thesis, Univ. of Washington, Seattle, WA, 1988.

42. Zhu, G. and Skelton, R.E., "Robust Properties of Periodic Discrete and Multirate Systems," *IEEE Trans. Auto. Control*, Vol. 36, No. 5, May 1992, p. 610-615.

APPENDIX A. DESIGN RESULTS

Following are the state space matrices for the optimized flutter suppression system digital processors discussed in Section 3.0.

A.1. SINGLE-RATE 2^ND^ ORDER

STP=BTP=0.02 sec; *N*=1. See Section 3.4.2.1 for a description of the sampling schedule.

$$\begin{bmatrix}\bar{z}_1(m+1,0)\\ \bar{z}_2(m+1,0)\end{bmatrix}=\begin{bmatrix}0 & 1\\ -0.61542 & 1.3562\end{bmatrix}\begin{bmatrix}\bar{z}_1(m,0)\\ \bar{z}_2(m,0)\end{bmatrix}+\begin{bmatrix}0 & -0.87258\\ 1 & -0.94601\end{bmatrix}\begin{bmatrix}\text{TE Accel}(m,0)\\ \text{LE Accel}(m,0)\end{bmatrix}$$

$$\text{CS Cmd}(m,0)=10^{-5}\begin{bmatrix}2.8302 & -13.621\end{bmatrix}\begin{bmatrix}\bar{z}_1(m,0)\\ \bar{z}_2(m,0)\end{bmatrix}$$

A.2. MULTIRATE SUCCESSIVE LOOP CLOSURES

STP=0.02 sec; BTP=0.08 sec; *N*=4. See Section 3.4.2.2 for a description of the sampling schedule.

Update during first STP of the BTP:

$$\begin{bmatrix}\bar{z}_f(m,1)\\ \bar{z}_s(m+1,0)\end{bmatrix}=\begin{bmatrix}0.75673 & 0\\ 0 & -0.47672\end{bmatrix}\begin{bmatrix}\bar{z}_f(m,0)\\ \bar{z}_s(m,0)\end{bmatrix}+\begin{bmatrix}-1 & 0.37644\\ -10^{-4} & 0.53661\end{bmatrix}\begin{bmatrix}\text{TE Accel}(m,0)\\ \text{LE Accel}(m,0)\end{bmatrix}$$

$$\text{CS Cmd}(m,0)=10^{-4}\begin{bmatrix}2.35354 & -2.5338\end{bmatrix}\begin{bmatrix}\bar{z}_f(m,0)\\ \bar{z}_s(m,0)\end{bmatrix}$$

Update during second STP of the BTP:

$$\bar{z}_f(m,2)=0.75673\bar{z}_f(m,1)+\begin{bmatrix}-1 & 0.37644\end{bmatrix}\begin{bmatrix}\text{TE Accel}(m,1)\\ \text{LE Accel}(m,1)\end{bmatrix}$$

$$\text{CS Cmd}(m,1)=10^{-4}\begin{bmatrix}-2.5338 & 2.35354\end{bmatrix}\begin{bmatrix}\bar{z}_f(m,1)\\ \bar{z}_s(m,0)\end{bmatrix}$$

Update during third STP of the BTP:

$$\bar{z}_f(m,3)=0.75673\bar{z}_f(m,2)+\begin{bmatrix}-1 & 0.37644\end{bmatrix}\begin{bmatrix}\text{TE Accel}(m,2)\\ \text{LE Accel}(m,2)\end{bmatrix}$$

$$\text{CS Cmd}(m,2)=10^{-4}\begin{bmatrix}-2.5338 & 2.35354\end{bmatrix}\begin{bmatrix}\bar{z}_f(m,2)\\ \bar{z}_s(m,0)\end{bmatrix}$$

Update during fourth STP of the BTP:

$$\bar{z}_f(m+1,0) = 0.75673\bar{z}_f(m,3) + \begin{bmatrix} -1 & 0.37644 \end{bmatrix} \begin{bmatrix} \text{TE Accel}(m,3) \\ \text{LE Accel}(m,3) \end{bmatrix}$$

$$\text{CS Cmd}(m,3) = 10^{-4} \begin{bmatrix} -2.5338 & 2.35354 \end{bmatrix} \begin{bmatrix} \bar{z}_f(m,3) \\ \bar{z}_s(m,0) \end{bmatrix}$$

We assumed that $\bar{z}_S$ is updated during the first STP of the BTP, but it it can be updated during any STP of the BTP.

A.3. MULTIRATE MULTIPLEXED INPUT

STP=0.02 sec; BTP=0.04 sec; N=2. See Section 3.4.2.3 for a description of the sampling schedule

Update during first STP of the BTP: Only the TE Accelerometer is sampled. The LE Accel value is held from the previous STP.

$$\begin{bmatrix} \bar{z}_1(m,1) \\ \bar{z}_2(m,1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.14712 & 0.88072 \end{bmatrix} \begin{bmatrix} \bar{z}_1(m,0) \\ \bar{z}_2(m,0) \end{bmatrix} + \begin{bmatrix} -1.3322 & 186.76 \\ -0.75421 & 136.42 \end{bmatrix} \begin{bmatrix} \text{TE Accel}(m,0) \\ \text{LE Accel}(m-1,1) \end{bmatrix}$$

$$\text{CS Cmd}(m,0) = 10^{-5} \begin{bmatrix} 8.6277 & -8.7583 \end{bmatrix} \begin{bmatrix} \bar{z}_1(m,0) \\ \bar{z}_2(m,0) \end{bmatrix}$$

Update during second STP of the BTP: Only the LE Accelerometer is sampled.The TE Accel value is held from the previous STP.

$$\begin{bmatrix} \bar{z}_1(m+1,0) \\ \bar{z}_2(m+1,0) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2.3304 & 3.7275 \end{bmatrix} \begin{bmatrix} \bar{z}_1(m,1) \\ \bar{z}_2(m,1) \end{bmatrix} + \begin{bmatrix} -2.5371 & -191.09 \\ -0.28724 & -189.04 \end{bmatrix} \begin{bmatrix} \text{TE Accel}(m,0) \\ \text{LE Accel}(m,1) \end{bmatrix}$$

$$\text{CS Cmd}(m,1) = 10^{-4} \begin{bmatrix} 3.7645 & -4.6910 \end{bmatrix} \begin{bmatrix} \bar{z}_1(m,1) \\ \bar{z}_2(m,1) \end{bmatrix}$$

A.4. SINGLE-RATE FAULT TOLERANT

STP=0.005 sec; BTP=0.005 sec; N=1. See Section 3.4.2.4 for a description of the sampling schedule.

$$\begin{bmatrix} \bar{z}_1(m+1,0) \\ \bar{z}_2(m+1,0) \\ \bar{z}_3(m+1,0) \\ \bar{z}_4(m+1,0) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.48177 & 2.4151 & -4.3750 & 3.4415 \end{bmatrix} \begin{bmatrix} \bar{z}_1(m,0) \\ \bar{z}_2(m,0) \\ \bar{z}_3(m,0) \\ \bar{z}_4(m,0) \end{bmatrix}$$

$$+10^{-5} \begin{bmatrix} 4.2073 & 6.4437 \\ -0.06264 & 1.3823 \\ -2.1575 & -1.1393 \\ -3.1600 & -2.3865 \end{bmatrix} \begin{bmatrix} \text{TE Accel}(m,0) \\ \text{LE Accel}(m,0) \end{bmatrix}$$

$$\text{CS Cmd}(m,0) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{z}_1(m,0) \\ \bar{z}_2(m,0) \\ \bar{z}_3(m,0) \\ \bar{z}_4(m,0) \end{bmatrix}$$

APPENDIX B. M-FILES USED TO DEFINE THE FLUTTEk SUPPRESSION SYSTEM SYNTHESIS PROBLEM

B.1. PAPA ABCD

Format: **[am, bm, cm, dm, vm]=PAPAabcd(fname, rolloff, form)**

Description: Creates state space matrices defining the PAPA wing at operating point specified in **fname** such that

$$\dot{x} = \mathbf{am}x + \mathbf{bm}u$$
$$y = \mathbf{cm}x + \mathbf{dm}u$$

$$\text{where } y = \left\{ \begin{array}{l} \text{plunge} \\ \text{pitch} \\ \text{plunge rate} \\ \text{pitch rate} \\ \text{TE accelerometer} \\ \text{LE accelerometer} \\ \text{command to actuator} \\ \text{CS control surface} \\ \text{CS control surface rate} \\ \text{CS control surface accel} \\ \text{mode 1} \\ \text{mode 2} \\ \text{mode 3} \\ \text{mode 4} \end{array} \right\} \text{ and } \quad u = \left\{ \begin{array}{l} \text{CS command} \\ \text{Dryden filter input} \end{array} \right\}$$

Inputs: **fname** text variable containing the name of the operating point of interest. e.g. 'freon_m5_q75'. **fname** must have the same name as the file which contains the data

rolloff frequency in rad/sec of first order anti-aliasing roll-off at the sensors. The filter has the form

$$y_{filtered} = \frac{\mathbf{rolloff}}{s+\mathbf{rolloff}} y_{unfiltered}$$

form indicates the desired form

if **form** = 0 : **am, bm, cm, dm** is unchanged from original data

1 : **am, bm, cm, dm** is block diagonal

2 : **am, bm, cm, dm** is block diagonal with scaled states and outputs

Outputs: **am, bm, cm, dm** state space description of the plant

vm transformation matrix used to obtain modal form

B.2. FSSCOMP

Format: **[cmp, sz, su, sy, stp, stppbtp]=FSScomp(ctype)**

Description: Generates the digital processor gain matrices and sampling schedule description for the four compensator described in Section 3.

Inputs: **ctype** specifies the desired compensator

if **ctype** = 'sr' then **FSScmp** returns a description of the 2^{nd} order Single-Rate design

'mrslc' then **FSScmp** returns a description of the Multirate Successive Loop Closure design

'mrmi' then **FSScmp** returns a description of the Multirate w/ Multiplexed Input design

'srft then **FSScmp** returns a description of the Single-Rate Fault Tolerant design

Outputs: **cmp, sz, su, sy, stp, stppbtp** a description of the compensator used by the synthesis algorithm. See Attachment 4.

B.3. MROPT_SR OR MRMI

Format: **mropt_srORmrmi**

Description: Defines the input data for the 2^{nd} order single-rate compensator or multirate compensator with multiplexed inputs. The user needs to comment and uncomment three lines to switch between the SR and the MRMI design. These are indicated in the text of the script.

Inputs: none

Outputs: Outputs to global variabies used by optimization routine and defined in Section 3.3 of Attachment 4

B.4. MROPT_MRSLC

Format: **mropt_mrslc**

Description: Defines the input data for the multirate compensator with successive loop closure fonn.

Inputs: none

Outputs: Outputs to global variables used by optimization routine and defined in Section 3.3 of Attachment 4

B.5. MROPT_SRFT

Format: **mropt_srft**

Description: Defines the input data for the single-rate fault tolerant compensator.

Inputs: none

Outputs: Outputs to global variables used by optimization routine and defined in Section 3.3 of Attachment 4

ATTACHMENT 1.
REDUCED ORDER MULTIRATE COMPENSATOR SYNTHESIS

A reprint of:

Mason, G.S. and Berg, M.C., "Reduced Order Multirate Compensator Synthesis," *AIAA Jour. Guid. Contr. and Dynamics*, Vol 15, No. 3 May-June 1992, pp. 700-706.

JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 15, No. 3, May–June 1992

Reduced-Order Multirate Compensator Synthesis

Gregory S. Mason and Martin C. Berg
University of Washington, Seattle, Washington 98195

A method for synthesizing reduced-order multirate compensators is presented. Necessary conditions for which the compensator parameter values minimize an infinite time quadratic cost function are derived. An algorithm for finding compensator parameter values which satisfy the necessary conditions is described. This algorithm is then used to design several tip position controllers for a two-link robot arm.

Introduction

IN many cases, a multirate compensator can provide better performance than a single-rate compensator requiring the same number of real-time computations. Berg, for example, was able to reduce the steady-state rms response of states and controls to a disturbance for a simple mass-spring-mass system nearly 20% by using a multirate compensator over a single-rate compensator.[1] Numerous other examples have been provided in the literature by Berg,[1-3] Amit,[4,5] and Yang.[6] Although multirate compensators can provide improved performance over single-rate compensators, they are also, in general, more complicated to design.

The complexity of multirate compensators stems from the fact that they are by nature time varying, periodically time varying for most practical applications. Not only must designers choose multiple sampling/update rates for the compensator, but they must also determine the parameter values for a time-varying compensator.

One method for designing multirate compensators is multirate linear quadratic Gaussian (LQG).[4] Multirate LQG is the multirate equivalent of single-rate LQG and is straightforward to solve because the equations governing the solution are similar to those for the single-rate case. Multirate LQG, however, results in a full-order compensator which has periodically time-varying gains. For many applications full-order, time-varying compensators are not practical.

A generalized algorithm for multirate synthesis (GAMS)[6] was developed by Yang to overcome many of the shortcomings of multirate LQG. Yang's algorithm can synthesize reduced-order multirate compensators with or without time-varying gains by using a numerical gradient-type search to find optimum compensator parameter values. His algorithm uses a finite time cost function in its problem formulation, unlike multirate and single-rate LQG which use an infinite time cost function. By using a finite time cost function, Yang's algorithm eliminates the numerical problem that arises when a destabilizing compensator is encountered during the numerical search. Even though Yang's algorithm uses a closed-form expression for the gradient, the calculations necessary to perform the gradient-type search are extremely cumbersome.

In this paper, we present a new algorithm for synthesizing reduced-order multirate compensators with or without time-varying gains. The algorithm utilizes the compensator structure of Yang's algorithm, but the problem is formulated using an infinite time, instead of a finite time, cost function. This allows us to derive necessary conditions for which the multirate compensator minimizes the cost function. The equations for the necessary conditions are fairly simple and can be solved directly using a standard nonlinear equation solver, eliminating many of the numerical complexities of Yang's algorithm.

General Multirate Compensator

Before deriving the equations governing a reduced-order multirate compensator, we will first present the structure for a general multirate compensator. We restrict our discussion for

Gregory S. Mason was born in Spokane, Washington, in 1961. He received the B.S. degree in mechanical engineering from Gonzaga University in 1983 and the M.S. degree in Computer Integrated Manufacturing from Georgia Institute of Technology in 1984. From 1984 to 1988 he worked in the robotics lab at the Naval Undersea Warfare Engineering Station in Keyport, Washington. Currently he is pursuing the Ph.D. degree in mechanical engineering at the University of Washington. His research interests include digital controls, robotics, and flexible automation.

Martin C. Berg was born in Everett, Washington, in 1952. He received the B.S. and M.S. degrees from the University of Washington in 1975 and 1978, respectively, and the Ph.D. degree in mechanical engineering from Stanford University in 1986. From 1978 to 1980 he worked in flight controls research for the Boeing Commercial Airplane Company. In 1986 he joined the faculty at the University of Washington, where he is currently an Assistant Professor of Mechanical Engineering. His research interests concern the design of robust sampled-data control systems.

Received Nov. 8, 1990; revision received March 3, 1991; accepted for publication April 8, 1991.

now to compensators with time-invariant gains and sampling/update rates whose ratios are rational numbers.

A general multirate compensator is shown in Fig. 1. Each input y, output u, and state $\bar{z}$ is sampled/updated at a rate which, in general, represents the desired bandwidth of the input or output with which it is associated. The variable $\bar{y}$ is the value of y currently available to the digital processor from the zero-order hold; while $\bar{u}$ is the current output from the digital processor which is held with a zero-order hold to form the output u. When the sampling/update rates have ratios which are rational numbers, the sampling/update schedule is periodically time varying. We define the greatest common divisor of all of the sampling/update periods as the shortest time period (*STP*) and the least common multiple of all of the sampling/update periods as the basic time period (*BTP*) (see Fig. 2).

The state equations for the multirate compensator pictured in Fig. 1 are

$$\begin{Bmatrix}\bar{z}\\ \bar{y}\\ \bar{u}\end{Bmatrix}_{k+1}=\begin{bmatrix}[I-s_{z,k}]+s_{z,k}\bar{A} & s_{z,k}\bar{B}[I-s_{y,k}] & 0\\ 0 & [I-s_{y,k}] & 0\\ s_{u,k}\bar{C} & s_{u,k}\bar{D}[I-s_{y,k}] & [I-s_{u,k}]\end{bmatrix}$$

$$\times\begin{Bmatrix}\bar{z}\\ \bar{y}\\ \bar{u}\end{Bmatrix}_{k}+\begin{bmatrix}s_{z,k}\bar{B}s_{y,k}\\ s_{y,k}\\ s_{u,k}\bar{D}s_{y,k}\end{bmatrix}y_k \tag{1}$$

$$u_k=\begin{bmatrix}s_{u,k}\bar{C} & s_{u,k}\bar{D}[I-s_{y,k}] & [I-s_{u,k}]\end{bmatrix}\begin{Bmatrix}\bar{z}\\ \bar{y}\\ \bar{u}\end{Bmatrix}_{k}$$

$$+[s_{u,k}\bar{D}s_{y,k}]y_k \tag{2}$$

where $\bar{u}$ is a hold state used to model the sampler and zero-order hold between $\bar{u}$ and u. The $s_{y,k}$, $s_{z,k}$, and $s_{u,k}$ are switching matrices for y, $\bar{z}$, and u, respectively, that model the system's sampling/update activity at the start of the kth *STP*. Also, $s_{*,k}$ has the form

$$s_{*,k}=\begin{bmatrix}r_1 & 0 & 0 & & 0\\ 0 & r_2 & 0 & & 0\\ \vdots & & & & \vdots\\ 0 & & 0 & r_{m_*-1} & 0\\ 0 & & 0 & 0 & r_{m_*}\end{bmatrix}$$

where

$$r_j=\begin{cases}1 & \text{if the } j\text{th ``*'' } (\bar{z}, y, \text{ or } u) \text{ is sampled/updated at the start of the } k\text{th } STP\\ 0 & \text{otherwise}\end{cases}$$

m_z = the number of states ($\bar{z}$)

m_y = the number of inputs (y)

m_u = the number of outputs (u)

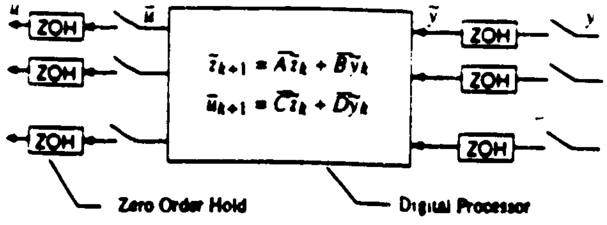

u
ZOH
ū
ZOH
ZOH
ỹ
y
ZOH
ZOH
ZOH
$\bar{z}_{k+1} = \bar{A}\bar{z}_k + \bar{B}\bar{y}_k$
$\bar{u}_{k+1} = \bar{C}\bar{z}_k + \bar{D}\bar{y}_k$
Zero Order Hold
Digital Processor

Fig. 1 A general multirate compensator.

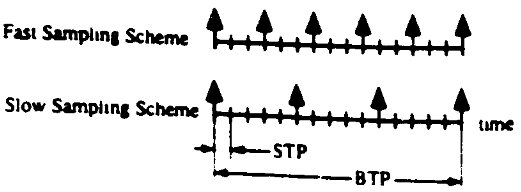


Fig. 2 Example of a multirate sampling scheme.

A more complete discussion of this compensator structure can be found in Refs. 6 and 7.

Equations (1) and (2) can be written more compactly as

$$z_{k+1}=A_kz_k+B_ky_k \tag{3}$$

$$u_k=C_kz_k+D_ky_k \tag{4}$$

where

$$z_k\equiv\begin{Bmatrix}\bar{z}\\ \bar{y}\\ \bar{u}\end{Bmatrix}_k$$

Equations (3) and (4) form a single-rate periodically time-varying system with a sampling rate of one *STP* and a period of one *BTP*. If $N = BTP/STP$, then $A_k = A_{k+N}$, $B_k = B_{k+N}$, $C_k = C_{k+N}$, and $D_k = D_{k+N}$.

Even though A_k, B_k, C_k, and D_k are periodically time varying, the multirate compensator gains, $\bar{A}$, $\bar{B}$, $\bar{C}$, and $\bar{D}$, are time invariant. The periodicity of the multirate compensator is due to multirate sampling/updating, not the compensator gains. In the remainder of this section, we will demonstrate how the time-invariant compensator gains, $\bar{A}$, $\bar{B}$, $\bar{C}$, and $\bar{D}$, can be separated from the periodic compensator matrices A_k, B_k, C_k, and D_k.

Define the composite compensator matrix as

$$P_k=\begin{bmatrix}D_k & C_k\\ B_k & A_k\end{bmatrix} \tag{5}$$

and factor P_k as follows:

$$P_k=S_{1k}\bar{P}S_{2k}+S_{3k} \tag{6}$$

where

$$\bar{P}=\begin{bmatrix}\bar{D} & \bar{C}\\ \bar{B} & \bar{A}\end{bmatrix} \tag{7}$$

$$S_{1k}=\begin{bmatrix}s_{u,k} & 0\\ 0 & s_{z,k}\\ 0 & 0\\ s_{u,k} & 0\end{bmatrix} \tag{8}$$

$$S_{2k}=\begin{bmatrix}s_{y,k} & 0 & I-s_{y,k} & 0\\ 0 & I & 0 & 0\end{bmatrix} \tag{9}$$

$$S_{3k}=\begin{bmatrix}0 & 0 & 0 & I-s_{u,k}\\ 0 & I-s_{z,k} & 0 & 0\\ s_{y,k} & 0 & I-s_{y,k} & 0\\ 0 & 0 & 0 & I-s_{u,k}\end{bmatrix} \tag{10}$$

Equation (6) is a key result. It allows us to factor the time-invariant compensator gains, the unknown parameters we will solve for in the next section, out of the time-varying compensator.

It is important to note the difference between P_k and $\bar{P}$ in Eq. (6). P_k (with a subscript) is a periodically time-varying matrix defined by Eq. (5). It includes all of the information about the compensator gains and the sampling/update schedule. $\bar{P}$ is a constant matrix which contains only the gains for the compensator. P_k can be written in terms of $\bar{P}$ and S_{1k}, S_{2k}, and S_{3k} using Eq. (6). S_{1k}, S_{2k}, and S_{3k} are periodically time-varying matrices which contain a description of the sampling/update scheme.

Derivation of the Necessary Conditions

In this section, we will use the results of the previous section to derive the necessary conditions for the reduced-order multirate compensator. The multirate problem to be solved is as follows.

Given:

the discretized plant model

$$\hat{x}_{k+1} = \hat{F}\hat{x}_k + \hat{G}\hat{u}_k + \hat{W}w_k \tag{11}$$

$$\hat{y}_k = \hat{H}\hat{x}_k + v_k \tag{12}$$

where $\hat{F}$, $\hat{G}$, $\hat{W}$, and $\hat{H}$ are obtained by discretizing the analog plant matrices at one *STP*; w_k and v_k are discrete-time Gaussian white noise inputs; $\hat{u}$ is the control input from the compensator; and $\hat{y}$ is the sampled sensor output.

Find:

the multirate control law with a prescribed dynamic order and sampling schedule, of the form of Eqs. (1) and (2) which minimizes a quadratic cost function of the form

$$J = \lim_{l\to\infty} \sum_{k=1}^{N} E\left\{\begin{Bmatrix} x_k \\ u_k \end{Bmatrix}^T \begin{bmatrix} Q_1 & M \\ M^T & Q_2 \end{bmatrix} \begin{Bmatrix} x_k \\ u_k \end{Bmatrix}\right\} \tag{13}$$

where E is the expected value operator, and the summation from 1 to N accounts for the fact that the closed-loop system is periodically time varying. A prescribed sampling schedule implies that the values of $s_{z,k}$, $s_{y,k}$ and $s_{u,k}$ are known.

Using Eqs. (3) and (4), it is easy to see that this problem is essentially a time-varying feedback problem—a time-invariant plant with a periodically time-varying compensator. One thing that makes this problem difficult is that the compensator has an explicit form, that of Eqs. (1) and (2), in which only certain parameters, $\bar{A}$, $\bar{B}$, $\bar{C}$, and $\bar{D}$, can be adjusted to minimize J.

To solve the multirate control problem, we cast it into output feedback form and follow a derivation similar to Mukhopadhyay's for the single rate case.[8,9] Using Eqs. (3) and (4) and Eqs. (11) and (12), we write the output feedback equations

$$\begin{Bmatrix} \hat{x}_{k+1} \\ z_{k+1} \end{Bmatrix} = \begin{bmatrix} \hat{F} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{x}_k \\ z_k \end{Bmatrix} + \begin{bmatrix} \hat{G} & 0 \\ 0 & I \end{bmatrix} \begin{Bmatrix} \hat{u}_k \\ z_{k+1} \end{Bmatrix} + \begin{bmatrix} \hat{W} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} w_k \\ v_k \end{Bmatrix} \tag{14}$$

$$\begin{Bmatrix} \hat{y}_k \\ z_k \end{Bmatrix} = \begin{bmatrix} \hat{H} & 0 \\ 0 & I \end{bmatrix} \begin{Bmatrix} \hat{x}_k \\ z_k \end{Bmatrix} + \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} w_k \\ v_k \end{Bmatrix} \tag{15}$$

$$\begin{Bmatrix} \hat{u}_k \\ z_{k+1} \end{Bmatrix} = \begin{bmatrix} D_k & C_k \\ B_k & A_k \end{bmatrix} \begin{Bmatrix} \hat{y}_k \\ z_k \end{Bmatrix} \tag{16}$$

Equations (14–16) can be written more compactly as

$$x_{k+1} = Fx_k + Gu_k + W\eta_k \tag{17}$$

$$y_k = Hx_k + V\eta_k \tag{18}$$

$$u_k = P_k y_k \tag{19}$$

It is important to keep in mind that P_k in Eq. (19) corresponds to the P_k in Eq. (5), a periodically time-varying matrix which contains all of the information about the multirate compensator gains and sampling/update rates.

The closed-loop system is

$$x_{k+1} = F_{ck}x_k + G_{ck}\eta_k \tag{20}$$

where

$$F_{ck} = F + GP_kH \tag{21}$$

$$G_{ck} = W + GP_kH \tag{22}$$

The state covariance propagation for this system obeys

$$X_{k+1} = F_{ck}X_kF_{ck}^T + G_{ck}RG_{ck}^T \tag{23}$$

where

$$X_k = E\{x_kx_k^T\}, \qquad R = E\{\eta_k\eta_k^T\}$$

Equations (20–22) represent a periodically time-varying system with a period of one *BTP*. We can generate a single-rate system by repeated application of Eq. (20) over one *BTP*.[10] The single-rate system can be written as

$$x_{k+N} = F_{bk}x_k + G_{bk}\eta_{bk} \tag{24}$$

where

$$F_{bk} = F_{c(k+N-1)}F_{c(k+N-2)}F_{c(k+N-3)}\cdots F_{ck} \tag{25}$$

$$G_{bk} = \left[F_{c(k+N-1)}F_{c(k+N-2)}\cdots F_{c(k+1)}G_{ck} \,\middle|\, \times F_{c(k+N-1)}F_{c(k+N-2)}\cdots F_{c(k+2)}G_{c(k+1)} \,\middle|\, \cdots \,\middle|\, G_{c(k+N-1)}\right] \tag{26}$$

$$\eta_{bk} = \begin{bmatrix} \eta_k \\ \eta_{k+1} \\ \vdots \\ \eta_{k+N-1} \end{bmatrix}$$

This single-rate system has exactly the same values for x as the periodically time-varying, closed-loop system at each *BTP*. However, the values of x at the intermediate *STP* are lost because x is incremented by N in Eq. (24) but only by 1 in Eq. (20). There are N such single-rate systems associated with Eq. (20). They can be written as

$$x_{k+N+i} = F_{b(k+i)}x_{k+i} + G_{b(k+i)}\eta_{k+i}, \quad \text{for } i = 1,2,\ldots,N \tag{27}$$

If F_{bk} is stable, then the periodically time-varying system Eq. (20) is stable.[11] We can calculate the steady-state covariance for x using the following Lyapunov equations:

$$X_k = F_{bk}X_kF_{bk}^T + G_{bk}R_bG_{bk}^T, \qquad \text{for } k = 1,2,\ldots,N \tag{28}$$

$$R_b = \begin{bmatrix} R & 0 & & 0 \\ 0 & R & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & 0 & R \end{bmatrix}$$

Note that X_k is periodic, that is it varies within one *BTP*, but from *BTP* to *BTP* $X_k = X_{k+N}$. Once we have calculated X_k at any k using Eq. (28), we can use Eq. (23) to propagate it over the *BTP*. This eliminates the need to solve Eq. (28) N times.

Now, using Eqs. (23) and (13), and the properties of the trace (Tr) operator, we can write the cost function for the stabilized system as (see Refs. 8 and 9)

$$J = \sum_{k=1}^{N} \mathrm{Tr}\Big\{\big[Q_1 + MP_kH + (MP_kH)^T + (P_kH)^TQ_2P_kH\big]X_k + (P_kV)^TQ_2P_kVR\Big\} \tag{29}$$

Adjoin the covariance constraints Eq. (23) to the cost J using Lagrange multipliers, Λ_k, to obtain

$$\mathcal{J} = \sum_{k=1}^{N} \mathrm{Tr}\Big\{\big[Q_1 + MP_kH + (MP_kH)^T + (P_kH)^TQ_2P_kH\big]X_k + (P_kV)^TQ_2P_kVR + \Lambda_{k+1}^T\big[F_{ck}X_kF_{ck}^T + G_{ck}RG_{ck}^T - X_{k+1}\big]\Big\} \tag{30}$$

with $X_1 = X_{N+1}$.

Necessary conditions for minimum J are

$$\frac{\partial \mathcal{J}}{\partial X_k} = 0, \qquad \frac{\partial \mathcal{J}}{\partial \Lambda_{k+1}} = 0, \qquad \frac{\partial \mathcal{J}}{\partial \tilde{P}} = 0 \tag{31}$$

In addition, $\partial^2\mathcal{J}/\partial\tilde{P}^2$ must be positive definite for a minimum J.

Substituting Eq. (30) into Eq. (31) and replacing P_k with $P_k = S_{1k}\tilde{P}S_{2k} + S_{3k}$ from Eq. (6), we obtain

$$\frac{\partial \mathcal{J}}{\partial X_k} = 0 = Q_1 + MP_kH + (MP_kH)^T + (P_kH)^TQ_2P_kH + F_{ck}^T\Lambda_{k+1}F_{ck} - \Lambda_k \tag{32}$$

for $k = 1,2,\ldots,N$ with $\Lambda_k = \Lambda_{k+N}$.

$$\frac{\partial \mathcal{J}}{\partial \Lambda_{k+1}} = 0 = F_{ck}X_kF_{ck}^T + G_{ck}RG_{ck}^T - X_{k+1} \tag{33}$$

for $k = 1,2,\ldots,N$ with $X_k = X_{k+N}$.

$$\frac{\partial \mathcal{J}}{\partial \tilde{P}} = 0 = 2\sum_{k=1}^{N} S_{1k}^T\Big\{\big[Q_2 + G^T\Lambda_{k+1}G\big]P_k\big[HX_kH^T + VRV^T\big] + \big[M^T + G^T\Lambda_{k+1}F\big]X_kH^T\Big\}S_{2k}^T \tag{34}$$

Equations (32-34) are a set of coupled matrix equations. They make up necessary conditions for $\tilde{P}$, which is comprised of the multirate compensator gain matrices $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$, in Eqs. (1) and (2), to minimize the cost function J. Values of $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$, found by solving Eqs. (32-34), can be substituted into Eqs. (1) and (2), along with the definition of the sampling schedule, $s_{c,k}$, $s_{u,k}$, and $s_{y,k}$, to form the complete time-varying multirate compensator.

To ensure that the compensator gains satisfying Eqs. (32-34) minimize J, we should also check that the Hessian of $\mathcal{J}$ with respect to $\tilde{P}$ is positive definite. Our present algorithm does not calculate the Hessian explicitly, but uses an approximate value calculated by the numerical search algorithm discussed in the next section.

Equations (32-34) were derived assuming time-invariant compensator gains. We can easily derive the corresponding equations for periodically time-varying gains. Let

$$\tilde{A} = \tilde{A}_k, \qquad \tilde{B} = \tilde{B}_k, \qquad \tilde{C} = \tilde{C}_k, \qquad \tilde{D} = \tilde{D}_k \tag{35}$$

with the restriction that $\tilde{A}_{k+N} = \tilde{A}_k$, $\tilde{B}_{k+N} = \tilde{B}_k$, $\tilde{C}_{k+N} = \tilde{C}_k$, and $\tilde{D}_{k+N} = \tilde{D}_k$. Define the composite periodically time-varying compensator matrix

$$\tilde{P}_k = \begin{bmatrix} \tilde{D}_k & \tilde{C}_k \\ \tilde{B}_k & \tilde{A}_k \end{bmatrix} \tag{36}$$

Then replace $\tilde{P}$ with $\tilde{P}_k$ in Eq. (30) and differentiate with respect to $\tilde{P}_k$ to obtain

$$\frac{\partial \mathcal{J}}{\partial \tilde{P}_k} = 0 = S_{1k}^T\Big\{\big[Q_2 + G^T\Lambda_{k+1}G\big]P_k\big[HX_kH^T + VRV^T\big] + \big[M^T + G^T\Lambda_{k+1}F\big]X_kH^T\Big\}S_{2k}^T \quad \text{for } k = 1,2,\ldots,N \tag{37}$$

Thus, for every new set of compensator gains we obtain one new equation of the form of Eq. (37).

Equations (32-34) are very similar to the single-rate equations. In fact, if we set S_{1k}, S_{2k}, and S_{3k} so that they correspond to a single-rate system, and $N = 1$, we obtain the exact results derived by Mukhopadhyay for the single-rate case.[8]

Implementation

To find a reduced-order multirate compensator that minimizes the cost function J, we need to solve Eqs. (32-34) for the compensator gains $\tilde{P}$. A flowchart of the algorithm used to determine the compensator gains is shown in Fig. 3. Using the prescribed sampling schedule the algorithm first discretizes the analog plant model, analog cost function, and analog process noise model. (See Ref. 2 for a discussion of the relevant discretization procedures.) Equations (32-34) are then solved for the compensator gains using a gradient-type search in Mat-

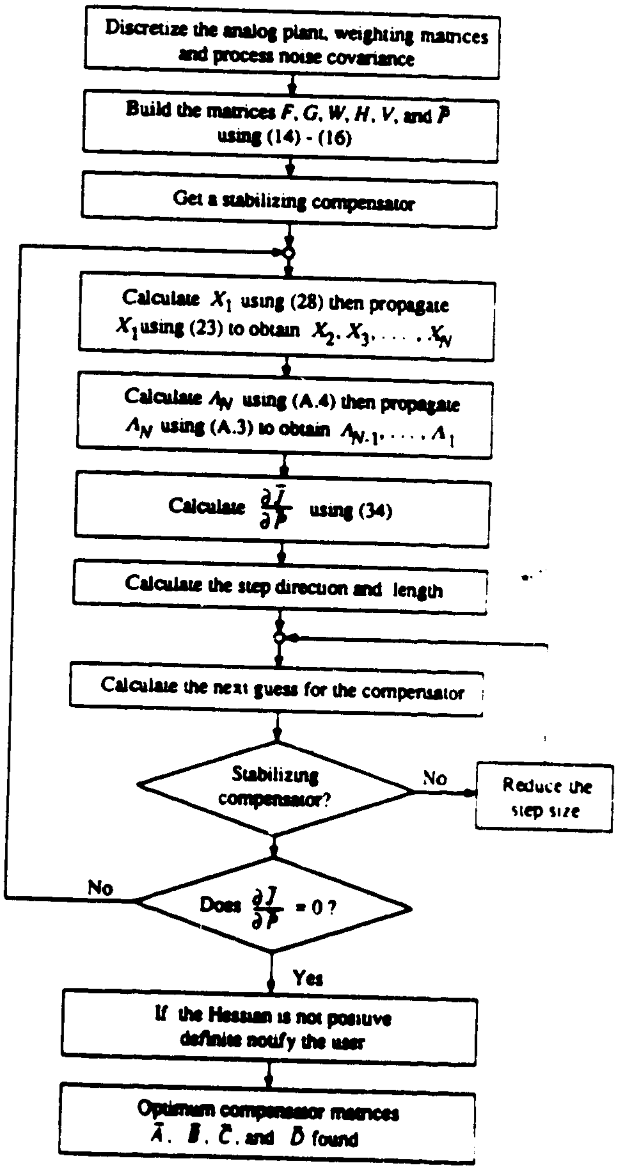
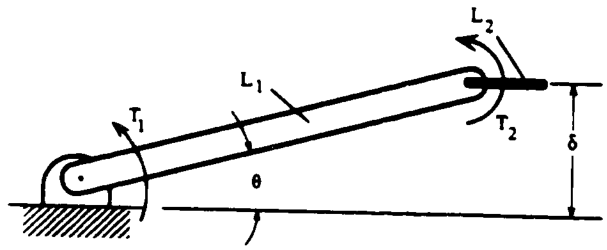

Discretize the analog plant, weighting matrices and process noise covariance
Build the matrices F, G, W, H, V, and $\tilde{P}$ using (14) - (16)
Get a stabilizing compensator
Calculate X_1 using (28) then propagate X_1 using (23) to obtain $X_2, X_3, \ldots, X_N$
Calculate Λ_N using (A.4) then propagate Λ_N using (A.3) to obtain $\Lambda_{N-1}, \ldots, \Lambda_1$
Calculate $\frac{\partial \bar{J}}{\partial \tilde{P}}$ using (34)
Calculate the step direction and length
Calculate the next guess for the compensator
Stabilizing compensator?
No
Reduce the step size
No
Does $\frac{\partial \mathcal{J}}{\partial \tilde{P}} = 0$?
Yes
If the Hessian is not positive definite notify the user
Optimum compensator matrices $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$ found

Fig. 3 Flowchart of optimization algorithm.



Parameters:

	Mass	Length
L$_1$	1.235 kg	0.965 m
L$_2$	0.163 kg	0.167 m

Inputs: Torque T$_1$ and T$_2$
Outputs: θ and δ

Fig. 4 Planar two-link robot arm.

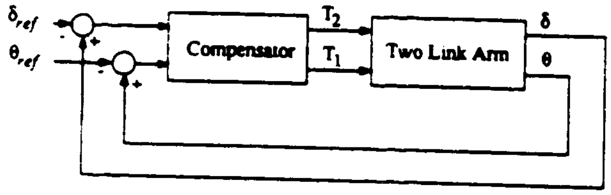


Fig. 5 TLA plant/compensator configuration.

lab.[12] We chose a gradient-type search to solve Eqs. (32-34) because it allows us to easily add constraints on the parameters values—simple equality constraints were used to find the optimized compensators in the next section. The equations necessary to solve for the Lagrange multipliers are located in the Appendix, Eqs. (A3) and (A4). To ensure that the solution represents a minimum J, the algorithm checks that the Hessian of J with respect to the free parameters in $\bar{P}$ is positive definite at the solution point.

Because Eqs. (32-34) are not valid when the closed-loop system is unstable, the algorithm 1) must be provided with an initial stabilizing compensator, and 2) must result in a stabilizing compensator at every iteration. From our experience, finding an initial stabilizing compensator is generally not a problem. Many systems suitable for multirate control can be stabilized using successive loop closure with minimal cross coupling between the control loops. A stabilizing multirate compensator can then be obtained by discretizing the individual continuous control loops at the desired sampling rates. When there are no constraints on its structure, a stabilizing compensator can also be obtained using the boot strapping method of Boussard.[13,14] For difficult multirate control problems, where a stabilizing compensator cannot be found using either of the preceeding two methods, one can always use Yang's algorithm to find a stabilizing compensator and then switch to our algorithm to complete the optimization. In our experience, Yang's algorithm usually converges to a stabilizing solution quickly—it is the optimization of the compensator parameters that is time consuming.

To avoid the problem of destabilizing compensators during the iteration process, we included a check in the algorithm which systematically reduces the step size to ensure that the compensator is stabilizing. Because the gradient of the cost function with respect to the compensator parameters becomes very large near the stability boundary, the algorithm is always forced away from a destabilizing solution as long as it never steps over the stability boundary into an unstable region.

Even though our algorithm was programmed as an interpreted Matlab M-File we found that it still performed better than Yang's algorithm which runs as compiled Fortran. The primary difference between the two algorithms is in the complexity of the expression for the gradient of J with respect to the compensator parameters. Calculation of the gradient expression for Yang's problem involves diagonalization of the closed-loop system and evaluation of several matrix equations with nested summations. Compare Eqs. (32-34) with Eqs. (112-115) in Ref. 3 to see the difference in the complexity of the two gradient expressions.

Table 1 Sampling/update rates for TLA

T_2	Sample/update rate, s
θ	0.225
δ	0.028125
T_1	0.225
T_2	0.028125

Two-Link Robot Arm Example

We used a mathematical model of a planar two-link robot arm (TLA) to demonstrate the capabilities of our algorithm. This is the same model used by Yang,[6] and so we were able to verify our results by direct comparison. A diagram of the TLA is shown in Fig. 4.

The goal of our design was to control the tip position δ of the arm via a multirate compensator. We used the following analog cost function and process noise covariance matrices from Ref. 6.

$$J = \lim_{t\to\infty} E\left\{ x^T \begin{bmatrix} 0.21 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 18.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + u^T \begin{bmatrix} 0.01 & 0 \\ 0 & 0.69444 \end{bmatrix} u \right\} \tag{38}$$

where

$$x = \begin{Bmatrix} \theta \\ \dot{\theta} \\ \delta \\ \dot{\delta} \end{Bmatrix}, \qquad u = \begin{Bmatrix} T_1 \\ T_2 \end{Bmatrix}$$

$$E\{ww^T\} = \begin{bmatrix} 0.69444 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{39}$$

We assumed perfect measurement and that plant disturbances enter the system coincident with the control torques. The sampling/update rates are given in Table 1.

Five difference compensators were designed: an analog LQR, a multirate lead/lead, an optimized multirate lead/lead, an optimized multirate general second order, and an optimized single-rate general second order. We used a smooth step input to δ_{ref} and θ_{ref} defined as follows:

$$\delta_{ref}(t) = \begin{cases} 0.005\left[1 - \cos\left(\dfrac{\pi t}{T_c}\right)\right] \text{m}, & t \le T_c \\ 0.001 \text{ m}, & t \ge T_c \end{cases}$$

$$\theta_{ref}(t) = \frac{\delta_{ref}(t)}{L_1 + L_2}, \qquad T_c = 0.125 \text{ s} \tag{40}$$

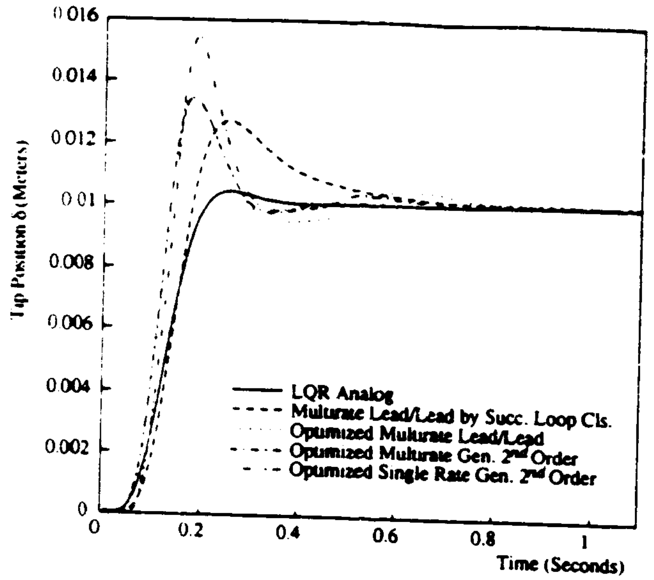


Fig. 6a Tip δ response to a smooth step command to tip position.

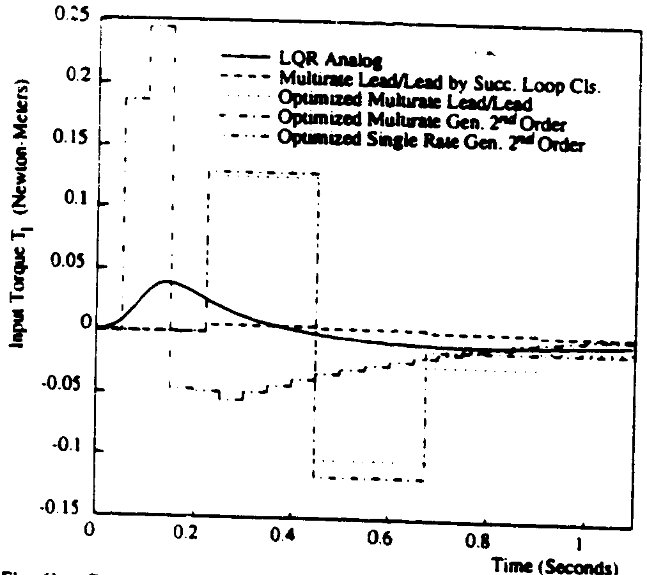


Fig. 6b Control torque T_1 response to a smooth step command to tip position.

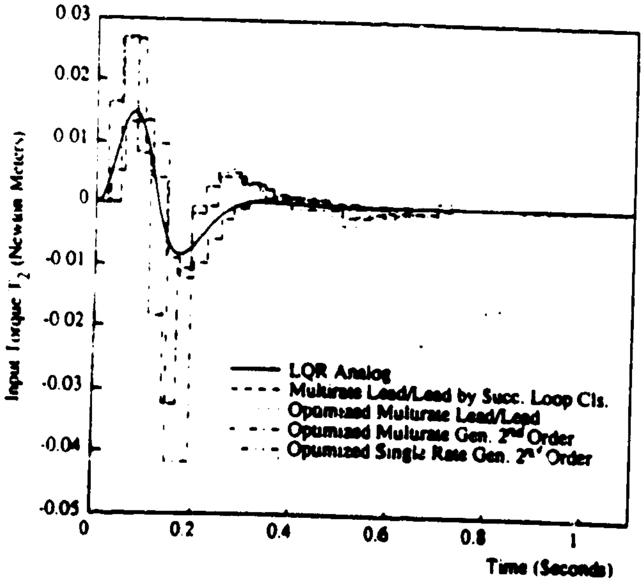


Fig. 6c Control torque T_2 response to a smooth step command to tip position.

and the servo configuration shown in Fig. 5 to measure the performance of the different compensators. The response of the TLA for the five compensators is shown in Figs. 6a-6c.

The analog LQR compensator used full state feedback. We provided this compensator as an example of the response possible using the cost function weighting matrices of Eq. (38).

The multirate lead/lead was found using successive loop closures. We designed the control loops in the discrete domain so that the eigenvalues of the closed-loop system matched those we obtained using LQR transformed to discrete time. This compensator consists of two simple lead loops: one from δ to T_2 operating at the fast sampling/update rate, and one from θ to T_1 operating at the slow sampling/update rate.

The final three compensators were synthesized using our new algorithm and the cost-weighting matrices used to design the analog LQR compensator. The optimized multirate lead/lead was found by optimizing the pole/zero locations and gains of the lead/lead compensator found by successive loop closures.

The optimized multirate general second-order compensator uses the same sampling/update scheme as the lead/lead compensators but has the compensator structure of Eq. (41), where a_{ij}, b_{ij}, c_{ij}, and d_{ij} are the parameters which were optimized. This compensator has the maximum number of independent free parameters possible for a second-order system.

$$\hat{A} = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 & b_{12} \\ b_{21} & 1 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, \quad \hat{D} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \tag{41}$$

The optimized single-rate general second-order compensator is a single-rate equivalent of the multirate general second-order compensator. It has the same structure as the multirate general second-order compensator, Eq. (41), but uses a single sampling rate. This sampling rate was chosen such that the number of computations required to implement either the multirate or single-rate compensators during real-time operation are the same.

Our results are the same as those obtained using Yang's algorithm. They demonstrate how multirate compensators can provide better performance than single-rate compensators by trading lower bandwidth control of the slow modes for higher bandwidth control of the fast modes. In this example, we were able to reduce the tip response overshoot 40% and the peak control torque 25% by using a multirate controller over a single-rate controller.

Conclusions

In this paper, we have presented a new algorithm for synthesizing reduced-order multirate compensators. It can be used to design compensators of arbitrary structure and dynamic order, with independent sampling/update rates for the compensator inputs, outputs, and states. This algorithm provides the versatility of Yang's algorithm without the numerical complexities associated with the finite time cost function.

Finally, we do not want to discount Yang's algorithm altogether because, while our algorithm requires an initial stabilizing compensator, Yang's does not. For those problems where finding an initial stabilizing compensator is difficult, we can always use Yang's algorithm to find a stabilizing compensator and then quickly optimize the compensator parameter values with our algorithm.

Appendix

Given a P_k which stabilizes the multirate system, we can calculate the steady-state values of Λ_k where Λ_k is defined by Eq. (32) rewritten here as Eq. (A1).

$$0 = Q_1 + MP_kH + (MP_kH)^T + (P_kH)^T Q_2 P_k H + F_{ck}^T \Lambda_{k+1} F_{ck} - \Lambda_k \tag{A1}$$

for $k = 1,2,\ldots,N$ with $\Lambda_k = \Lambda_{k+N}$.

First simplify Eq. (A1) by defining

$$Q_3 = \begin{bmatrix} Q_1 & M \\ M^T & Q_2 \end{bmatrix}, \qquad J_k = \begin{bmatrix} I \\ P_k H \end{bmatrix} \tag{A2}$$

where I is an identity matrix. Then Eq. (A1) can be written as

$$\Lambda_k = J_k^T Q_3 J_k + F_{ck}^T \Lambda_{k+1} F_{ck} \tag{A3}$$

for $k = 1,2,\ldots,N$ with $\Lambda_k = \Lambda_{k+N}$.

Equation (A3) represents a periodically time-varying Lyapunov equation. We can create an equivalent single-rate system by repeated application of Eq. (A3).

$$\Lambda_k = J_{dk}^T Q J_{dk} + F_{dk}^T \Lambda_k F_{dk} \tag{A4}$$

for $k = 1,2,\ldots,N$ with $\Lambda_k = \Lambda_{k+N}$.

$$F_{dk} = F_{c(k+N-1)} F_{c(k+N-2)} F_{c(k+N-3)} \cdots F_{ck} \tag{A5}$$

$$J_{dk} = \begin{bmatrix} J_{(k+N-1)} F_{c(k+N-2)} F_{c(k+N-3)} \cdots F_{ck} \\ J_{(k+N-2)} F_{c(k+N-3)} \cdots F_{ck} \\ \vdots \\ J_k \end{bmatrix} \tag{A6}$$

$$Q_d = \begin{bmatrix} Q_3 & 0 & \cdots & 0 \\ 0 & Q_3 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & Q_3 \end{bmatrix}$$

Equation (A4) is a time-invariant Lyapunov equation which can be solved for Λ_k. Once any Λ_k has been found, the propagation Eq. (A3) can be used to find the remaining Λ_k.

Acknowledgment

This research was supported by NASA Langley Research Grant NAG-1-1055.

References

[1] Berg, M. C., Amit, N., and Powell, J. D., "Multirate Digital Control System Design," *IEEE Transactions on Automatic Control*, Vol. AC-33, Dec. 1988, pp. 1139-1150.

[2] Berg, M. C., "Design of Multirate Digital Control Systems," Ph.D. Thesis, Stanford Univ., Stanford, CA, 1986.

[3] Berg, M. C., and Yang, G. S., "A New Algorithm for Multirate Digital Control Law Synthesis," *Proceedings of the 27th IEEE Conference on Decision Control*, Institute of Electrical and Electronics Engineering, Piscataway, NJ, Dec. 1988, pp. 1685-1690.

[4] Amit, N., "Optimal Control of Multirate Digital Control Systems," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Stanford Univ., Stanford, CA, Rept. 523, 1980.

[5] Amit, N., and Powell, J. D., "Optimal Control of Multirate Systems," AIAA Paper 81-1797; see also *Proceedings of the AIAA Guidance and Control Conference*, AIAA, New York, 1981.

[6] Yang, G. S., "A Generalized Synthesis Method for Multirate Feedback Control Systems," Ph.D. Thesis, Univ. of Washington, Seattle, WA, 1988.

[7] Berg, M. D., Mason, G. S., and Yang, G. S., "A New Multirate Sampled-Data Control Law Structure and Synthesis Algorithm," *1991 American Control Conference*, Boston, MA, June 1991, pp. 2749-2754.

[8] Mukhopadhyay, V., "Digital Robust Control Law Synthesis Using Constrained Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 2, 1989, pp. 175-181.

[9] Mukhopadhyay, V., Newsom, J. R., and Abel, I., "A Method for Obtaining Reduced-Order Control Laws for High-Order Systems Using Optimization" NASA TP 1876, Aug. 1981.

[10] Meyer, R. A., and Burrus, C. S., "A Unified Analysis of Multirate and Periodically Time-Varying Digital Filters," *IEEE Transactions on Circuits in Systems*, Vol. CAS-22, No. 3, 1975, pp. 162-168.

[11] Kono, M., "Eigenvalue Assignment in Linear Periodic Discrete-Time Systems," *International Journal of Control*, Vol. 32, No. 1, 1980, pp. 149-158.

[12] *Pro-Matlab Users Guide*, Mathworks Inc., South Natick, MA, 1989.

[13] Broussard, J. R., and Halyo N., "Optimal Multirate Output Feedback," *Proceedings of the 23rd Conference on Decision and Control*, Institute of Electrical and Electronics Engineers, Piscataway, NJ, Dec. 1984, pp. 926-929.

[14] Halyo, N., and Brousard, J. R., "A Convergent Algorithm for the Stocastic Infinite-Time Discrete Optimal Output Feedback Problem," *Proceedings of the 1981 Joint Automatic Control Conference*, Vol. 1, Charlottesville, VA, American Automatic Controls Council, Section WA-1E.

ATTACHMENT 2.
ROBUSTNESS ANALYSIS OF A MULTIRATE FLUTTER SUPPRESSION SYSTEM

A reprint of:

Mason, G.S. and Berg, M.C., "Robustness Analysis of a Multirate Flutter Suppression System," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 16, No. 5, 1993, pp. 992-926.

JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 16, No. 5, September–October 1993

Robustness Analysis of a Multirate Flutter Suppression System

Gregory S. Mason* and Martin C. Berg†
University of Washington, Seattle, Washington 98195

A unified approach for analyzing the robustness and performance of multirate systems is presented. The approach is a compilation of existing multirate performance and robustness analysis results, along with some new results, all presented in a consistent state-space formulation. The approach is used to analyze the robustness of a multirate flutter suppression system designed for a model wing. The example highlights some of the practical considerations in multirate performance and robustness analysis.

Introduction

NUMEROUS approaches have been taken to analyze the stability and robustness of multirate systems. Most notable is the work of Thompson.[1] Thompson and Dailey,[2] Meyer and Burrus,[3] and Khargonekar et al.[4] References 1 and 2 analyzed the gain and phase margins of a multirate system using an approach based on Kranc vector switch decomposition. Meyer and Burrus studied the stability of multirate systems and their frequency domain responses by applying the concept of block processing to multirate systems. Khargonekar et al. analyzed the robustness of periodic compensators, a super set of multirate compensators, using isomorphisms.

Although these approaches seem quite diverse, fundamentally they are very similar. These three approaches, along with most multirate analysis techniques, use a transformation, such as Kranc vectors or block processing, to convert the multirate system into an equivalent single-rate system which can be analyzed with established single-rate techniques. The single-rate results are then used to characterize the stability and robustness of the original multirate system.

In this paper we compile the important multirate analysis results from Refs. 1–6 and present them in a unifying state-space formulation. In addition, we provide some new results that clarify the relationship between a multirate system and its single-rate equivalent. Finally, we apply all of these results to a practical example: a multirate flutter suppression system designed for a model wing.

Summary of Multirate Analysis Tools

In this section we summarize some important and useful multirate robustness analysis results. These results are applicable to multirate systems that are linear, causal, finite-dimensional, and whose sample/update/delay activities are periodic and synchronized to a common clock.

Result 1

A multirate system can be modeled as an equivalent single-rate system (ESRS). Modeling a multirate system as an ESRS is fundamental to multirate robustness analysis. The ESRS allows one to manipulate and analyze a multirate system as if it were single rate. Using the ESRS, single-rate and multirate systems can be combined in series or in feedback loops just as in classical control.[4] It has also been shown that a single-rate/multirate system will be stable whenever its ESRS is stable.[7]

The ESRS of a multirate system can be obtained by modeling the multirate system as a periodically time-varying system,[8,9] and transforming the periodically time-varying system into an ESRS.[3] The state-space representation of the ESRS is

$$x(m+1, 0) = A_E x(m, 0) + B_E u_E(m, 0) \tag{1a}$$

$$y_E(m, 0) = C_E x(m, 0) + D_E u_E(m, 0) \tag{1b}$$

where

$$y_E(m, 0) = \begin{bmatrix} y(m, 0) \\ y(m, 1) \\ \vdots \\ y(m, N-1) \end{bmatrix} \qquad u_E(m, 0) = \begin{bmatrix} u(m, 0) \\ u(m, 1) \\ \vdots \\ u(m, N-1) \end{bmatrix} \tag{2}$$

for $m = 0, 1, 2, \ldots,$ and $n = 0, 1, 2, \ldots, N-1$.

In these equations, $u(m, n)$ and $y(m, n)$ represent the values of the input and output, respectively, of the original multirate system at the $(mN+n)$th sampling instant. The integer N is the ratio of the least common multiple of all of the multirate system's sample/update/delay periods to their greatest common divisor. The subscript E denotes vectors and matrices strictly associated with the ESRS.

A key feature of an ESRS is that its input/output vectors are composite vectors containing the input/output values of the multirate system at N separate sampling times. Consequently, an ESRS is always multiple input, multiple output (MIMO) even if the original multirate system is single input, single output (SISO). Another key feature is that an ESRS always has a nonzero direct feed through term D_E. This is because D_E contains information about how past inputs affect the current output. For systems with no dynamics, the direct feed through term D_E is block diagonal. For example, the ESRS of a constant uncertainty matrix Δ is

$$\Delta_E = \text{block diag}[\Delta, \Delta, \ldots, \Delta] \tag{3}$$

with N blocks.

Refer to Refs. 3, 6, 8, and 9 for the details on modeling a multirate system as an ESRS.

Result 2

A discrete signal $w(m, n)$ is related to its ESRS signal $w_E(m, 0)$ as follows:

$$w(m, n) = [W_N(n)\; W_N(n-1) \cdots W_N(n-N+1)]\, w_E(m, 0)$$

where $W_N(n)$ is a switching function defined as

$$W_N(k) = \frac{1}{N}\sum_{l=0}^{N-1} \cos\frac{2\pi}{N}lk \tag{4a}$$

Received July 29, 1991; presented as Paper 91-2812 at the AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, Aug. 12–14, 1991; revision received Oct. 22, 1992; accepted for publication Oct. 22, 1992.

*Post Doctoral Researcher, Mechanical Engineering Department.

†Assistant Professor, Mechanical Engineering Department. Member AIAA.

so that

$$W_N(k) = 1 \qquad \text{if } k = \pm mN \qquad \text{for } m = 0, 1, \ldots$$
$$= 0 \qquad \text{otherwise} \tag{4b}$$

Result 2 provides a convenient mathematical connection between the input/output vectors of a multirate system and its ESRS. The form of W_N given in Eq. (4a) is useful for analytic results whereas that given in Eq. (4b) is convenient for numerical simulation.

Result 2 allows one to use single-rate analysis techniques to obtain results for multirate systems. For example, to compute the time domain response of a multirate system to a generic input, we can find the time domain response of the ESRS using any applicable single-rate technique, and then transform that response back to the original multirate system using result 2.

Result 3

The two-norm and rms value of a discrete signal $w(m, n)$ and its ESRS signal $w_E(m, 0)$ are equal or, equivalently,

$$\|w(m, n)\|_2 = \|w_E(m, 0)\|_2 \quad \text{and} \quad \text{rms}[w(m, n)] = \text{rms}[w_E(m, 0)]$$

Result 3 follows directly from Eq. (2) and the definition of the two-norm and rms value.[10]

Result 4

The maximum rms gain of a multirate system is given by the H-infinity norm of its ESRS transfer function $G_E(z)$ or, equivalently,

$$\sup_{\text{rms}(u) \neq 0} \frac{\text{rms}[y(m, n)]}{\text{rms}[u(m, n)]} = \|G_E(z)\|_\infty$$

It is well known that the maximum rms gain of a SISO single-rate system is equivalent to the maximum gain on that system's Bode plot. Although a transfer function for a multirate system does not exist in the traditional sense, we can see from result 4 that the H-infinity norm of a SISO multirate system plays the same role as the maximum Bode plot gain of a single-rate system.

Result 4 follows from result 3 and Refs. 10–12. It is also derived, in part, in Ref. 4.

Result 5

The singular values of a single-rate transfer function $G(z)$ and its ESRS transfer function $G_E(z)$ are related as follows:

$$\sigma[G_E(e^{jN\theta})] = \{\sigma[G(e^{j\theta})]^T \, \sigma[G(e^{j[\theta+(2\pi/N)]})]^T$$
$$\sigma[G(e^{j[\theta+(4\pi/N)]})]^T \cdots \sigma[G(e^{j[\theta+[2\pi(N-1)/N]]})]^T\}^T$$

where $\sigma[\cdot]$ denotes a column vector of singular values.

Result 5 relates the singular values of a single-rate system to the singular values of its ESRS and provides some insight into the significance of the singular values associated with the ESRS of a multirate system. From result 5 we can see that one effect of transforming $G(z)$ into $G_E(z)$ is that the singular values of $G(z)$ at high frequencies are aliased into $G_E(z)$ at lower frequencies. Consequently, the $u(m, n)$ in result 4 that results in the maximum rms gain does not necessarily contain the frequency ω associated with $\|G_E(e^{j\omega T})\|_\infty$ (T is the sampling period of the ESRS). The input signal of maximum rms gain must be constructed using the right singular vectors of $G_E(z)$ and result 2. We will demonstrate this procedure in the following section. (See the Appendix for a derivation of result 5.)

Result 6

The stability, gain margin, and phase margin of a SISO multirate system can be determined directly from a Nyquist plot of its ESRS. Recall from result 1 that the ESRS of a SISO multirate system is MIMO. Therefore, the multiloop Nyquist stability criterion must be used in result 6.[13] This, however, is one case where gain and phase margins taken from a multiloop Nyquist plot can be interpreted in the traditional sense because gain and phase variations at the multirate system's input/output apply simultaneously to all the inputs/outputs of the ESRS.

Result 6 follows from Eq. (3) and is derived in Ref. 1.

Result 7

The robustness of a multirate system can be determined by applying structured and unstructured singular value analysis to that system's ESRS. Given the ESRS transfer function G_E of the nominal system and the uncertainty transfer function Δ_E, we can apply established singular value analysis techniques to find the size of the smallest uncertainty $\bar{\sigma}(\Delta_E)$ that destabilizes the closed-loop system in Fig. 1. This result, however, is only a conservative estimate of the size of the actual smallest destabilizing uncertainty Δ. The input/output vectors of an ESRS are composite vectors, containing the input/output values of the multirate system at N sample times. Thus $\bar{\sigma}(\Delta_E)$ found using unstructured singular value analysis accounts for not only the fictitious perturbations normally associated with unstructured singular values, but also for time-varying and noncausal perturbations. A valid perturbation for a given $\bar{\sigma}(\Delta_E)$ might, for example, include block diagonal elements in Δ_E that are unequal. This corresponds to a time-varying perturbation because the gain between $u(m, n)$ and $y(m, n)$ varies with n. Another valid perturbation could include nonzero upper block diagonal elements in Δ_E. This corresponds to a noncausal perturbation because a future input $u(m, n+1)$ can affect the current output $y(m, n)$.

For the ESRS uncertainty Δ_E to represent the actual uncertainty Δ, its structure must obey Eq. (1). Finding $\bar{\sigma}(\Delta_E)$ subject to Eq. (1) requires the solution of a structured singular value problem. Unfortunately, even simply structured dynamic uncertainties in a single-rate/multirate system transform to uncertainties with complex structures in the ESRS. The complex structure makes it difficult to obtain a good estimate of the size of the smallest destabilizing structured perturbation. However, when the single-rate/multirate uncertainty is a constant, as is the case for many problems, the ESRS uncertainty is also a constant with a repeated block diagonal form [see Eq. (3)]. A good estimate of the solution of such a structured singular value problem with repeated blocks can be found using one of the methods in Refs. 14–17.

Result 7 follows directly from the fact that a single-rate/multirate system is stable if and only if its ESRS is stable.[7]

Application

In this section we apply the results of the previous section to a real world example: a multirate flutter suppression system for a model wing. This application points out some of the practicalities of multirate robustness analysis.

The model wing used in this example is being developed under the Benchmark Active Controls Project at the NASA Langley Research Center. It consists of a rigid airfoil mounted on a pitch and plunge apparatus (PAPA). The PAPA mount provides the two degrees of freedom needed to model classical wing flutter. The wing has one control surface located on the trailing edge (TE) of the airfoil. Two accelerometers measure pitch and plunge accelerations. We used a 15th-order mathematical model of the wing for the control system design. This model incorporates a second-order Dryden gust filter, a third-order actuator model, and a 10th-order

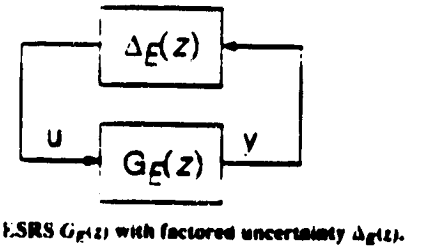


Fig. 1 ESRS $G_E(z)$ with factored uncertainty $\Delta_E(z)$.

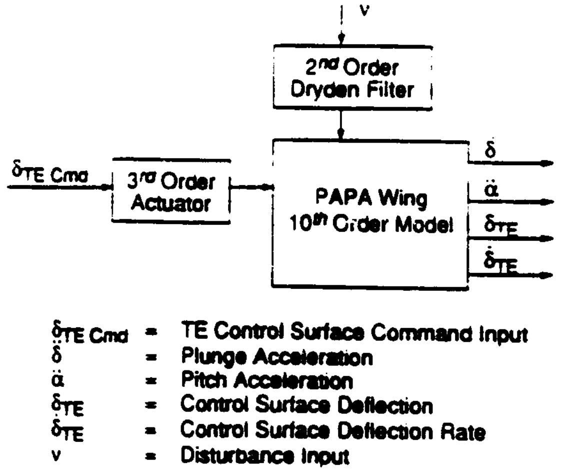

v
2nd Order Dryden Filter
δTE Cmd
3rd Order Actuator
PAPA Wing 10th Order Model
δ̈
α̈
δTE
δ̇TE
δTE Cmd = TE Control Surface Command Input
δ̈ = Plunge Acceleration
α̈ = Pitch Acceleration
δTE = Control Surface Deflection
δ̇TE = Control Surface Deflection Rate
v = Disturbance Input

Fig. 2 Block diagram of model wing with pitch and plunge apparatus.

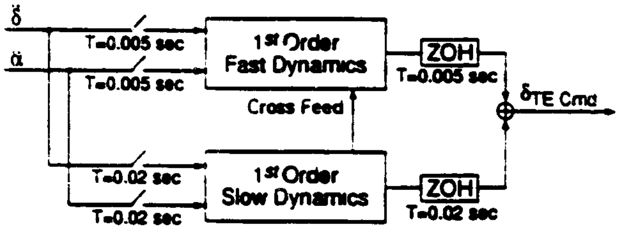


Fig. 3 Block diagram of the second-order multirate compensator.

airfoil model (two rigid body modes and six unsteady aero states). A block diagram of this model is shown in Fig. 2.

A second-order multirate flutter suppression system was designed for the model wing. The multirate compensator, shown in Fig. 3, consists of two first-order control loops. The slow loop is sampled/updated at 50 Hz. The fast loop is sampled/updated at 200 Hz, resulting in $N = 4$. We optimized all of the free parameters in this compensator using the multirate compensator synthesis algorithm described in Ref. 9. The free parameters were the pole and zero locations for the fast and slow loops, the gain values for the fast and slow loops, and gain values for the cross feed between the fast and slow loops.

To analyze the robustness of the closed-loop system we examined the gain and phase margins at the plant input and output and the rms gains from disturbance input to the control surface deflection and deflection rate. Gain and phase margins provide a measure of the uncertainty allowed in the plant model. The rms gains provide a measure of the allowable disturbance level before the control surface actuator limits are exceeded.

Traditional gain and phase margins at the plant input were calculated using result 6. The location of the corresponding complex loop gain k_u is shown in Fig. 4. The multiloop Nyquist diagram for the open-loop ESRS is shown in Fig. 5. Gain and phase margins calculated from the Nyquist diagram are given in Table 1.

Generalized gain and phase margins at the plant output were calculated using the ESRS and the structured singular value. For this analysis, the closed-loop system was cast into the form of the block diagram shown in Fig. 1. The uncertainty block Δ and the corresponding Δ_E are

$$\Delta = \begin{bmatrix} k_1 & 0 \\ 0 & k_1 \end{bmatrix} \qquad \Delta_E = \begin{bmatrix} I_N k_1 & 0 \\ 0 & I_N k_2 \end{bmatrix} \tag{5}$$

where I_N is an $N \times N$ identity matrix. The complex gain uncertainties k_1 and k_2 represent additive plant output uncertainties. They are shown in Fig. 4. G_E (in Fig. 1) is the nominal closed-loop system comprised of the model wing and the multirate flutter suppression system.

The form of the uncertainty Δ_E in Eq. (5) leads directly to a structured singular value problem with repeated scalar block uncertainties. Typically, an exact solution for such a problem is difficult to find. Fortunately, this problem has only two free parameters, k_1 and k_2, and an exact solution can be found. We calculated an exact value of the size of the smallest destabilizing uncertainty using the following equation[14]:

$$\frac{1}{\bar{\sigma}(\Delta_{\min})} = \sup_{0<\phi<\pi} \max_{0<\theta<2\pi} \rho\{U(\theta)G_E(e^{j\phi})\} \tag{6}$$

with

$$U(\theta) = \begin{bmatrix} I & 0 \\ 0 & e^{j\theta}I \end{bmatrix}$$

where $\bar{\sigma}(\Delta_{\min})$ is the maximum singular value of the smallest destabilizing Δ; and $\rho(\cdot)$ is the spectral radius. The resulting value of $\bar{\sigma}(\Delta_{\min})$ is given in Table 1.

Generalized gain and phase margins corresponding to $\bar{\sigma}(\Delta_{\min})$ were calculated using the method described in Ref. 18. Figure 6 shows the "region of guaranteed stability" for simultaneous (independent) gain and phase changes in $1+k_1$ and $1+k_2$.

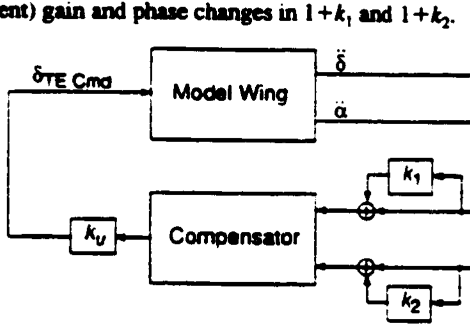


Fig. 4 Location of the complex gains used for gain and phase margin analysis; nominal system: $k_u = 1$, $k_1 = k_2 = 0$.

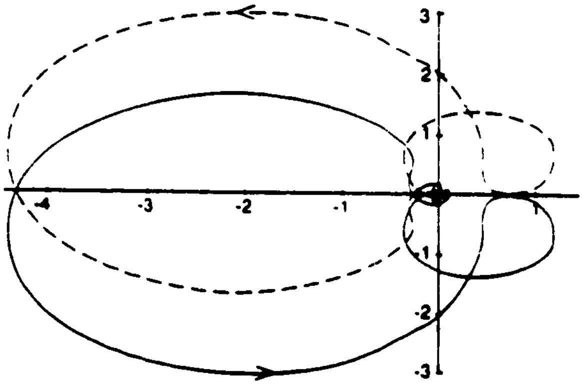


Fig. 5 Multiloop Nyquist plot of PAPA wing with multirate compensator; the control loop is broken at k_u with $k_1 = k_2 = 0$; open-loop system has two unstable poles.

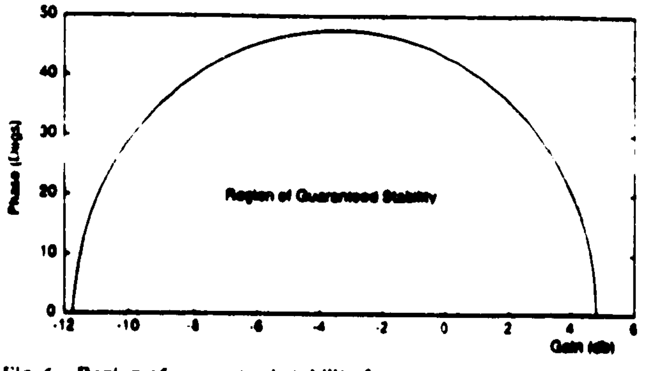


Fig. 6 Region of guaranteed stability for simultaneous changes in the gain and phase of $(1+k_1)$ and $(1+k_2)$.

Table 1 Robustness results for multirate flutter suppression system

Gain margin at plant input (with $k_1 = k_2 = 0$), dB	[−12, +10]
Phase margin at plant input (with $k_1 = k_2 = 0$), deg	±65
$\bar{\sigma}(\Delta_{\min})$ with $k_4 = 1$	0.74
Maximum rms gain, v to δ_{TE}, deg·s/in.	0.125
Maximum rms gain, v to $\dot{\delta}_{TE}$, deg/in.	4.67

The maximum rms gain values for disturbance input to control surface deflection and deflection rate were calculated using result 4. Result 4, however, is directly applicable only to discrete systems and not to our mixed continuous/discrete system. Therefore, we discretized the plant disturbance input and control surface deflection and deflection rate outputs at 500 Hz, assuming a zero-order hold at the disturbance input. This sampling rate is 30 times the Dryden filter roll-off frequency. The values of the maximum rms gains are then the peak values of the maximum singular value plots of this new multirate system (see Fig. 7). The peak values, which occur at $\omega_{\max} = 27$ rad/s, are given in Table 1. Since we assumed a stair step disturbance input, the values in Table 1 are lower, yet very close, bounds on the maximum rms gains.[19]

Contrary to what occurs in linear time invariant single-rate systems, we cannot assume that the signal producing the maximum rms gain $u_{\max\,\mathrm{rms}}(m, n)$ has the simple form $\sin(w_{\max}Tm)$. Instead, we must construct $u_{\max\,\mathrm{rms}}(m, n)$ using result 2. For our example, the input signal of maximum rms gain $u_{\max\,\mathrm{rms}}(m, n)$ from the disturbance v to the control surface deflection δ_{TE} was found as follows. Let

$$G_E(e^{j\omega_{\max}T}) = \text{transfer function from } v \text{ to } \delta_{TE} \text{ evaluated at } \omega_{\max} \qquad (7a)$$

and

$$\begin{bmatrix} \alpha_1 e^{j\phi_1} \\ \alpha_2 e^{j\phi_2} \\ \vdots \\ \alpha_V e^{j\phi_V} \end{bmatrix} = \text{right singular vector associated with } \sigma\,[G_E(e^{j\omega_{\max}T})] \qquad (7b)$$

Then, $u_{\max\,\mathrm{rms}}(m, n)$ is given by

$$u_{\max\,\mathrm{rms}}(m, n) = [W_V(n)\, W_V(n-1) \ldots W_V(n-N+1)] \times \begin{bmatrix} \alpha_1 \sin(\omega_{\max}Tm + \phi_1) \\ \alpha_2 \sin(\omega_{\max}Tm + \phi_2) \\ \vdots \\ \alpha_V \sin(\omega_{\max}Tm + \phi_V) \end{bmatrix} \qquad (8)$$

where W_V is the switching function described in result 2, and T is the sampling period of the ESRS.

From Eqs. (4) and (8) it is straightforward to see that, in general, the signal of maximum rms gain for a multirate system is comprised of the sum of sinusoids of several distinct frequencies. In this example, though, $u_{\max\,\mathrm{rms}}(m, n)$ is comprised almost purely of a single sinusoid of frequency of 27 rad/s. This is because higher frequency signals which might interact with the multirate compensator and increase the rms gain are attenuated by the Dryden filter and plant dynamics.

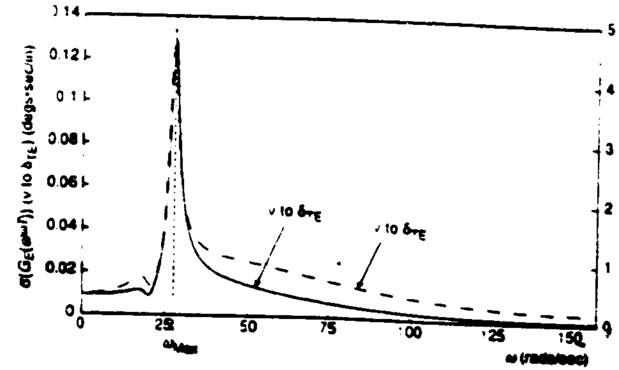


Fig. 7 Maximum singular value plots of v to δ_{TE} and v to $\dot{\delta}_{TE}$, $=0.02$ s.

Conclusions

In this paper we have summarized some important multirate robustness analysis results. The foundation for these results is the equivalent single-rate system. This system allows one to analyze the stability and robustness of a multirate system using well-known single-rate techniques.

There are, however, drawbacks to using the equivalent single-rate system. First, the composite structure of its inputs and outputs leads to matrix transfer functions of high order. This is a problem especially when the ratio of the longest to the shortest sampling/update/delay period of the multirate system is large. Second, even when the simplest constant parameter uncertainty model is used for robustness analysis, the composite input and output structure gives rise to a structured singular value problem with repeated diagonal blocks. This is a numerically difficult problem to solve.

In short, multirate system stability and robustness analysis based on the equivalent single-rate system is straightforward because the analysis can be performed using established single-rate techniques, but the results must be interpreted carefully in accordance with its input/output structure.

Appendix

Lemma:

$$\sigma[G_E(e^{jN\theta})] = \{\sigma[G(e^{j\theta})]^T\, \sigma[G(e^{j(\theta+2\pi/N)})]^T\, \sigma[G(e^{j(\theta+4\pi/N)})]^T \cdots \sigma[G(e^{j(\theta+(2\pi(N-1)/N))})]^T\}^T$$

Proof: Let $G(z)$ and $G_E(z^N)$ be the transfer functions of a single-rate system and its ESRS, respectively, such that

$$y(z) = G(z)u(z) \qquad \text{and} \qquad y_E(z^N) = G_E(z^N)u_E(z^N) \qquad (A1)$$

We have written $G_E(z^N)$ as a function of z^N because the sampling period for the ESRS is N times that of the single-rate system associated with $G(z)$.

It is shown in Ref. 3 that

$$y(z) = [I\;\; z^{-1}I\;\; z^{-2}I\; \cdots\; z^{-(N-1)}I\,]\, y_E(z^N) \qquad (A2)$$

where I is an identity matrix of appropriate dimensions.

Now define

$$\tilde{y}(z) = \begin{bmatrix} y(\phi^0 z) \\ y(\phi^1 z) \\ \vdots \\ y(\phi^{N-1} z) \end{bmatrix}, \qquad \tilde{u}(z) = \begin{bmatrix} u(\phi^0 z) \\ u(\phi^1 z) \\ \vdots \\ u(\phi^{N-1} z) \end{bmatrix} \qquad (A3)$$

and

$$\tilde{G}(z) = \text{block diag}[G(\phi^0 z), G(\phi^1 z), \ldots, G(\phi^{N-1} z)] \qquad (A4)$$

where

$$\phi = e^{(2\pi j/N)}$$

such that

$$\tilde{y}(z) = \tilde{G}(z)\tilde{u}(z) \tag{A5}$$

We can now write $\tilde{y}(z)$ and $\tilde{u}(z)$ in terms of $y_E(z^N)$ and $u_E(z^N)$ using Eq. (A2).

$$\tilde{y}(z) = T(z)y_E(z^N) \qquad \tilde{u}(z) = T(z)u_E(z^N) \tag{A6}$$

where

$$T(z) = \begin{bmatrix} I & z^{-1}I & \dots & z^{-(N-1)}I \\ I & (\phi^1 z)^{-1}I & & (\phi^1 z)^{-(N-1)}I \\ \vdots & & & \vdots \\ I & (\phi^{N-1}z)^{-1}I & \dots & (\phi^{N-1}z)^{-(N-1)}I \end{bmatrix} \tag{A7}$$

Then from Eqs. (A1) and (A5) through (A6)

$$G_E(z^N) = T^{-1}\tilde{G}(z)T \tag{A8}$$

Noticing that $T(e^{j\theta})T(e^{j\theta})^* = NI$, we can define a unitary matrix $\tilde{T}(e^{j\theta})$ such that

$$T(e^{j\theta}) = \sqrt{N}\tilde{T}(e^{j\theta}) \qquad \text{and} \qquad T(e^{j\theta})^{-1} = (1/\sqrt{N})\tilde{T}(e^{j\theta})^* \tag{A9}$$

Now, using Eqs. (A8) and (A9) and the properties of unitary matrices, it is straightforward to show that

$$\sigma[G_E(e^{jN\theta})] = \sigma[\tilde{G}(e^{jN\theta})] \tag{A10}$$

Since $\tilde{G}$ is block diagonal [see Eq. (A4)], and the singular values of a block diagonal matrix are the union of the singular values of each block, we can rewrite Eq. (A10) as follows:

$$\sigma[G_E(e^{jN\theta})] = \{\sigma[G(e^{j\theta})]^T \;\; \sigma[G(e^{j[\theta+(2\pi/N)]})]^T$$
$$\sigma[G(e^{j[\theta+(4\pi/N)]})]^T \cdots \sigma[G(e^{j[\theta+[2\pi(N-1)/N]]})]^T\}^T$$

Acknowledgment

This research was supported by NASA Langley Research Grants NCC1-156 and NAG-1-1055.

References

[1] Thompson, P. M., "Gain and Phase Margins of Multirate Sampled-Data Feedback Systems," *International Journal of Control*, Vol. 44, No. 3, 1986, pp. 833–846.

[2] Thompson, P. M., and Dailey, R. L., "Kranc Vector Switch Decomposition in State Space," Caltech Internal Rept., California Inst. of Technology, Davis, CA, Sept. 1985.

[3] Meyer, R. A., and Burrus, C. S., "A Unified Analysis of Multirate and Periodically Time-Varying Digital Filters," *IEEE Transactions on Circuits in Systems*, Vol. CAS-22, No. 3, March 1975, pp. 162–168.

[4] Khargonekar, P. P., Poolla, K., and Tannenbaum, A., "Robust Control of Linear Time-Invariant Plants Using Periodic Compensators," *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 11, Nov. 1985, pp. 1088–1096.

[5] Boykin, W. H., and Frazier, B. D., "Multirate Sampled-Data Systems Analysis Via Vector Operators," *IEEE Transactions on Automatic Control*, Vol. AC-20, No. 4, Aug. 1975, pp. 548–551.

[6] Apostolakis, I. S., and Jordan, D., "Multirate System Performance Evaluation Using Singular Value Analysis," *Proceedings of the 1990 American Control Conference*, IEEE, Piscataway, NJ, 1990, pp. 1502–1507.

[7] Kono, M., "Eigenvalue Assignment in Linear Periodic Discrete-Time Systems," *International Journal of Control*, Vol. 32, No. 1, 1980, pp. 149–158.

[8] Berg, M. C., Mason, G. S., and Yang, G. S., "A New Multirate Sampled-Data Control Law Structure and Synthesis Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 5, 1992, pp. 1183–1191.

[9] Mason, G. S., and Berg, M. C., "Reduced Order Multirate Compensator Synthesis," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, 1992, pp. 700–706.

[10] Boyd, S., and Doyle, J., "Comparison of Peak and RMS Gains for Discrete-Time Systems," *Systems and Control Letters*, Vol. 9, 1987, pp. 1–6.

[11] Maciejowski, J. M., *Multivariable Feedback Design*, Addison-Wesley, Reading, MA, 1990.

[12] Francis, B. A., "A Course in H_∞ Control Theory," *Lecture Notes in Control and Information Sciences*, Vol. 88, Springer-Verlag, New York, 1987.

[13] MacFarlane, A. G. J., "Return-Difference and Return-Ratio Matrices and Their Use in Analysis and Design of Multivariable Feedback Control Systems," *IEE Proceedings on Control Theory and Applications*, Pt. D, Vol. 117, No. 10, Oct. 1970, pp. 2037–2049.

[14] Doyle, J., "Analysis of Feedback Systems with Structured Uncertainties," *IEE Proceedings on Control Theory and Applications*, Pt. D, Vol 129, No. 6, Nov. 1982, pp. 243–250.

[15] Safonov, M.G., "Stability Margins of Diagonally Perturbed Multivariable Feedback Systems," *IEE Proceedings on Control Theory and Applications*, Pt. D, Vol. 129, No. 6, Nov. 1982, pp. 251–256.

[16] Safonov, M., and Doyle, J. C., "Minimizing Conservativeness of Robustness Singular Values," *Multivariable Control: New Concepts and Tools*, D. Reidel, Hingham, MA, 1984, pp. 197–207.

[17] Packard, A., and Doyle, J., "Structured Singular Value with Repeated Scalar Blocks," *Proceedings of the 1988 American Control Conference*, IEEE, Piscataway, NJ, 1988, pp. 1213–1218.

[18] Mukhopadhyay, V., and Newsom, J. R., "Multiloop System Stability Margin Study Using Matrix Singular Values," *Journal of Guidance, Control, and Dynamics*, Vol. 7, No. 5, 1984, pp. 582–587.

[19] Sivashankar, N., and Khargonekar, P. P., "Robust Stability and Performance Analysis of Sampled-Data Systems," *Proceedings of the IEEE Conference on Decision and Control*, IEEE Piscataway, NJ, pp. 881–885.

ATTACHMENT 3.
MULTIRATE FLUTTER SUPPRESSION SYSTEM FOR A MODEL WING

A reprint of:

Mason, G.S. and Berg, M.C., "Multirate Flutter Suppression System for a Model Wing," accepted for publication in the *AIAA Journal of Guidance, Control and Dynamics*.

Multirate Flutter Suppression System Design for a Model Wing

Gregory S. Mason[*]
Mechanical Engineering Department
Seattle University
Seattle, Washington 98122

Martin C. Berg[†]
Mechanical Engineering Department
University of Washington,
Seattle, Washington 98195

Research Supported by NASA Langley Research Grants NAG-1-1055 and NCC-1-156

ABSTRACT

A new methodology for multirate control system desig.ı iɜ described. It accommodates a general multiple-input multiple-output control law structure that allows the sampling rates for the plant sensor output signals, the update rates for the processor states, and the update rates for the plant control input signals to be independently specified. It includes a capability to design for multiple plant conditions so as to achieve robustness to plant parameter variations. Its analysis components include a method for determining conventional gain and phase margins, a method for determining a bound on the smallest destabilizing uncertainty, and a method for determining the maximum RMS gain of a multirate system. The methodology is demonstrated by application to the design of a multirate flutter suppression system for a model wing.

INTRODUCTION

Multirate control systems occur frequently in engineering practice. They have received comparatively little attention in the technical literature. There are several reasons for this. One is a lack of recognition by the research community of the practical motivations for multirate controllers. Compared to single-rate controllers, they offer the relatively obvious real-time computing efficiencies in multi-loop, multi-function, multi-time-scale systems. But with real-time computing hardware costs as low as they are, such efficiencies usually do not justify the additional complexity of a multirate design.

[*] Assistant Professor

[†] Associate Professor, Member AIAA

In practice, multirate controllers are often necessitated by hardware constraints. For example, when a sensor provides a signal that is updated only at a fixed interval, except when the update period happens to be a suitable sampling period for a single-rate controller, a multirate controller must be used.

The logistical burden that a multriate system presents is a second reason for the lack of attention given to multirate control systems. This burden is a consequence of the fact that a multirate system is time-varying from one sampling instant to the next. Fortunately, a well designed software package can spare the designer from most of the burden of the logistical difficulties, thereby allowing him or her to concentrate on the more fundamental design issues.

This paper describes a multirate control system design methodology for which we have developed such a software package. The methodology was originally proposed in Reference 1. It employs the control law synthesis algorithm described in Reference 2, and the modeling and analysis tools described in References 3 and 4. The description is via an application to the design of a multirate flutter suppression system for a model wing.

The remainder of the paper is divided into four sections. The first describes the model wing, its open-loop characteristics, and the flutter suppression system design goals. The second describes the design methodology and its application to the flutter suppression system design. The third presents the results of the flutter suppression system design. Conclusions are given the final section.

PROBLEM DESCRIPTION

The BACT Wing

The *Benchmark Active Controls Technology* (BACT) Wing is being developed at the NASA Langley Research Center to study the modeling, prediction, and control of aerodynamic flutter. It consists of a rigid airfoil mounted on a flexible base. The base, called the *Pitch and Plunge Apparatus* (PAPA), provides the two degrees of freedom necessary to model classical wing flutter. The airfoil has one control surface (CS) located on the trailing edge. Two accelerometers, one near the leading edge (LE) and one near the trailing edge (TE) measure the airfoil's motion. References 5-6 describe the BACT Wing in detail.

The flutter suppression system was designed using 16th order linear state models of the BACT Wing developed by NASA Langley's Structural Dynamics Division. Each model consists of 4 rigid body states corresponding to the pitch and plunge modes, 6 unsteady aerodynamic states, a second order actuator model, a second order Dryden filter, and two first order anti-aliasing filters. Figure 1 shows a block diagram of this structure. NASA provided us with 24 such models, each describing the dynamics of the wing in Freon at a different operating point. The operating points include dynamic pressures above and below the critical flutter pressure at three different mach numbers. See Table 1 for a summary of the operating points.

Table 1. Operating points for BACT Wing. All operating points assume Freon medium

	Dynamic Pressure (psf) (Unstable operating points are in gray)							
Mach 0.50	75*	100	122	**132**	**150**	**175**	**200**	**225***
Mach 0.70	75	100	125*	136	**146**	**175***	**200**	**225**
Mach 0.78	75*	100	125	141	**151**	**175**	**200**	**225***

* Operating points used for compensator synthesis

Open-Loop Characteristics of the BACT Wing

Two modes - pitch and plunge - dominate the open-loop dynamics of the BACT Wing model. For example, the poles and zeros of the CS command to the LE and TE accelerometer output transfer functions at mach 0.5 and 75 psf are shown in Figures 2(a) and 2(b). As dynamic pressure increases, one pair of these poles moves toward the right half plane and crosses the imaginary axis at the stability boundary. Figure 3 shows this pole movement. The corresponding movements of the open-loop poles not shown in Figure 3 are relatively small.

The dominant pitch and plunge modes are observable at all operating points with either the TE or the LE accelerometer outputs, and are controllable at all operating points using the CS command input. The zeros of the CS command to TE accelerometer and the CS command to LE accelerometer transfer functions are shown in Figure 2 for the mach 0.5 and 75 psf operating point. As dynamic pressure increases, the non-minimum phase zeros associated with the TE accelerometer migrate into the left half plane. The minimum phase zeros, associated with the LE accelerometer and located near the dominant poles, migrate into the right half plane. See Figure 3.

At low dynamic pressures the transfer functions from the CS command input to the TE and LE accelerometer outputs are non-minimum phase. Non-minimum phase systems are more difficult to control than minimum phase systems[7]. An alternative output is one which measures the difference between the two accelerometer signals. This new output is essentially pitch acceleration. The CS command to pitch acceleration transfer function is *minimum* phase for all operating points, and the BACT Wing is relatively easy to control using this new output. We chose not to use this output directly in our designs, however, because it does not adequately account for the inevitable uncertainty in the TE and LE acceleration measurements.

Design Objectives

Our primary objective was to design a multirate flutter suppression system that will stabilize the BACT Wing when it is flown (at some future date) in the wind tunnel at speeds between mach 0.5 and 0.78 and dynamic pressures between 75 psf and 225 psf. In addition, the following constraints, most of which are functions of the hardware that will be used to implement the control law, were specified by NASA.

Control Activity Constraint: For unity RMS white noise input disturbance (1 in/sec RMS), the steady state covariance of the CS deflection must not exceed 0.0625 deg^2 (0.25 deg RMS), and the CS deflection rate must not exceed 65 deg^2/sec^2 (8.0 deg/sec RMS).

Sampling Rate Restrictions: The minimum sampling period is 0.005 sec. For multirate sampling, all sampling periods must be multiples of 0.005 sec.

Computational Delay: All compensators must account for a minimum 0.005 sec computational delay.

Robustness Constraints: The gain and phase margins at the compensator output, which is a scalar signal, must be at least ±6db and ±45°. The maximum singular value of the smallest destabilizing multiplicative uncertainty at the compensator input must be 0.75, which corresponds to simultaneous gain and phase margins at the two sensor inputs to the compensator of ±6db and ±45° [8].

Finally, our multirate control law was to provide the same performance and stability robustness as a comparable single-rate controller yet require less hardware to implement.

THE DESIGN METHODOLOGY

We designed two flutter suppression systems for the BACT Wing: a single-rate system, for use as a baseline for comparison, and a multirate system. Each was designed using the methodology in Reference 1. This methodology defines a general approach and provides the specific tools needed to solve a design problem. The methodology has three parts: modeling the multirate system, optimizing the digital processor gains, and analyzing the performance and robustness of the closed-loop system. In the following paragraphs we describe the methodology and its application to the design of the two flutter suppression systems.

Modeling the Flutter Suppression System

A block diagram of a generic flutter suppression system is shown in Figure 1. In this system, each sampler at the plant output can operate at an independent rate, the digital processor can update each processor state at an independent rate, and each zero-order-hold at the compensator output can operate at an independent rate. This compensator model has the form of the Generalized Multirate Control Law Structure (GMCLS) discussed in References 1, 3 and 4. The GMCLS provides a framework for modeling multirate compensators and eliminates much of the bookkeeping involved with multiple sample/update rates and/or time delays. Using the GMCLS, it is straightforward to represent a multirate system as a periodically time-varying single-rate system. Later we will see that the resulting periodically time-varying system can be further transformed into a time invariant system by "lifting"[9] or "block processing"[10].

There are two components to the GMCLS: the "sampling schedule" and the "digital processor gains". The "sampling schedule" indicates the sequence of sample and update activities for all samplers, processor states, and zero-order-holds. In the GMCLS, all sample and update activities must occur at integer multiples of a specified time period T; and the sampling schedule must repeat itself every NT, where N is an integer. Often T and N are functions of the hardware used to implement the control law. The second component of the GMCLS is the "digital processor gains" A_z, B_z, C_z, and D_z. These gains determine the dynamics of the digital processor and are typically free design parameters.

We modeled both the single-rate and the multirate flutter suppression systems using the framework of the GMCLS. Both compensators have the form of the generic compensator shown in Figure 1, with two inputs, TE and LE accelerations, one output, CS command, and second order digital processor dynamics.

The single-rate compensator has a sample/update rate of 50 Hz, which is approximately 10 times the frequency of the dominant pitch and plunge modes. The state space structure of the compensator's digital processor is

$$\begin{Bmatrix} x_1(n+1) \\ x_2(n+1) \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \begin{Bmatrix} x_1(n) \\ x_2(n) \end{Bmatrix} + \begin{bmatrix} 0 & b_1 \\ 1 & b_2 \end{bmatrix} \begin{Bmatrix} \text{TE Accel}(n) \\ \text{LE Accel}(n) \end{Bmatrix} \qquad (1a)$$

$$\text{CScmd}(n) = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{Bmatrix} x_1(n) \\ x_2(n) \end{Bmatrix} \qquad (1b)$$

where x_1 and x_2 are the digital processor states; TE Accel and LE Accel are the acceleration inputs from the A/D converters; and CS cmd is the command output to the zero-order-hold. a_i, b_i, and c_i are the free gains to be optimized. The structure of (1) represents a minimal realization of a second order compensator[3].

The GMCLS sampling schedule corresponding to the single-rate compensator is shown in Figure 4. In this figure, circles on each time line indicate when a particular sample or update activity occurs. The GMCLS "digital processor gains" correspond to the matrix elements in the digital processor's state space description given in (1).

The multirate compensator was designed to provide the same performance and stability robustness as the single-rate compensator using a reduced number of analog-to-digital converters. In the multirate compensator, the digital processor states and CS output are updated at 50 Hz, while the accelerometers are sampled at 25 Hz. In addition, there is a 0.02 second delay between the sampling of the TE accelerometer output and the LE accelerometer output. Consequently, the multirate compensator requires only one A/D converter to sample both accelerometer outputs.

To maximize the benefits of its multiplexed sampling schedule, the multirate compensator uses periodically time-varying digital processor gains. One set of gains is used when sampling the TE accelerometei output and another set is used when sampling the LE accelerometer output. The state space structure of the multirate compensator's digital processor is thus

$$\begin{Bmatrix} x_1(n+1) \\ x_2(n+1) \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ a_1(n) & a_2(n) \end{bmatrix} \begin{Bmatrix} x_1(n) \\ x_2(n) \end{Bmatrix} + \begin{bmatrix} 0 & b_1(n) \\ 1 & b_2(n) \end{bmatrix} \begin{Bmatrix} \text{TE Accel}(n) \\ \text{LE Accel}(n) \end{Bmatrix} \tag{2a}$$

$$\text{CScmd}(n) = \begin{bmatrix} c_1(n) & c_2(n) \end{bmatrix} \begin{Bmatrix} x_1(n) \\ x_2(n) \end{Bmatrix} \tag{2b}$$

where x_1 and x_2 are the digital processor states; TE Accel and LE Accel are the acceleration inputs from the A/D converters; and CScmd is the command output to the zero-order-hold. $a_i(n)$, $b_i(n)$, and $c_i(n)$ are the free gains to be optimized. These gains are functions of n and are periodically time varying, e.g., $a_i(n) = a_i(n+2)$.

The sampling schedule for the multirate compensator is shown in Figure 5. Notice the multiplexed sampling scheme of the two A/D converters. The GMCLS "digital processor gains" correspond to the matrix elements in the digital processor's state space description given in (2), just as in the single-rate case.

Optimizing the Digital Processor Gains

To determine the values of the digital processor gains for the two compensators, we used the low order multirate compensator synthesis algorithm described in Reference. 2, along with the multiple plant conditions idea of Ly[11-12]. The synthesis algorithm uses numerical optimization to determine values of the digital processor gains that minimize a quadratic cost function. The multiple plant conditions idea employs a cost function which is the sum of the costs associated with a single compensator in feedback with a nominal plant and perturbed variations of that plant. The digital processor gains that minimize this new cost function are robust in that they stabilize the nominal plant and the specified variations of that plant.

The multiple plant conditions cost function has the form

$$J_m = \lim_{t\to\infty} \sum_{i=1}^{Np} E\left\{ \begin{Bmatrix} x_i(t) \\ u_i(t) \end{Bmatrix}^T \begin{bmatrix} Q_i & M_i \\ M_i^T & R_i \end{bmatrix} \begin{Bmatrix} x_i(t) \\ u_i(t) \end{Bmatrix} \right\} \tag{3}$$

where E is the expected value operator; the integer N_p is the number of simultaneous plant perturbations under consideration; the vectors x_i and u_i represent the plant states and control inputs, respectively, of the ith plant condition; and Q_i, M_i, and R_i are the weighting matrices associated with the ith plant condition, and are free parameters selected by the designer.

Using the synthesis algorithm in Reference 2 and the cost function in Eqn. (3), we found values of the digital processor gains for the single-rate and multriate flutter suppression systems that stabilized the BACT Wing at all 24 operating points. Instead of optimizing over all 24 operating points, however, we selected six representative ones. The six are indicated in Table 1.

For each of the six operating points, we selected a unique set of weights, Q_i, M_i, and R_i, for the cost function in (3). To select Q_i, M_i, and R_i we used that fact that the gains which minimize (3) corresponds to the LQ regulator feedback gains when the compensator is a continuous time design, the plant outputs are the values of the plant states, and the compensator is strictly a feedback gain. Accordingly, we first designed a continuous LQ regulator, one for each of the six operating points, that satisfied NASA's performance criterion. The weights were chosen so that the closed-loop damping of the pitch and plunge modes was greater than 0.07, and the RMS control surface activity constraints specified by NASA were satisfied. (For comparison, the damping in the open-loop BACT Wing at a stable dynamic pressure of 75 psf is approximately 0.025.) Next, we uniformly scaled Q_i, M_i, and R_i to obtain a unity LQ cost for each operating point for a 6 inch/sec RMS white noise disturbance input . Finally, we used the values of Q_i, M_i, and R_i from each LQ regulator design for the corresponding values in (3). Optimum values of the digital processor gains were found by minimizing the cost function (3) using the synthesis algorithm in Reference 2.

After optimization, we evaluated each compensator's performance and robustness using the methods discussed in the next section. One of the robustness measures we considered was the maximum singular value of the minimum destabilizing multiplicative uncertainty at the compensator inputs (a structured singular value). The size of the structured singular value for our initial designs was unacceptably small. It was less than 0.20 at some operating points, whereas NASA had specified a value of 0.75. To remedy this we added fictitious sensor noise and reoptimized the processor gains (our initial designs assumed no sensor noise). Our addition of fictitious sensor noise was motivated by the Loop Transfer Recover technique for LQG systems design[13].

Analyzing the Flutter Suppression Systems

Recall that we modeled the flutter suppression systems using the GMCLS. It is easy to represent a compensator having the GMCLS as a periodically time-varying single-rate system. Furthermore, it is straightforward to transform a single-rate periodically time-varying system into a single-rate time-invariant system using "lifting" or "block processing". We refer to the resulting single-rate time-invariant system as the Equivalent Single-Rate System (ESRS). See References 1 or 4 for a discussion of the properties of the ESRS.

An ESRS has the following form

$$x(k+N)=A_E x(k)+B_E u_E(k) \tag{4a}$$

$$y_E(k)=C_E x(k)+D_E u_E(k) \tag{4b}$$

where

$$y_E(k)=\begin{Bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+N-1) \end{Bmatrix} \text{ and } u_E(k)=\begin{Bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{Bmatrix} \tag{4c}$$

and where $x(k)$ are the states of the discrete system, $u(k)$ are the discrete inputs, and $y(k)$ are the sampled outputs. The subscript E denotes vectors and matrices strictly associated with the ESRS.

A key feature of an ESRS is that its input/output vectors are composite vectors containing the input/output values of the original system at N sample times. Consequently, an ESRS is always MIMO even if the original system is SISO. Another feature is that an ESRS always has a nonzero direct feed through term. When the original system has no dynamics, the direct feed through term of its ESRS, D_E, is block diagonal. For example, the ESRS of a constant matrix Δ is

$$\Delta_E = \text{Block Diag}[\Delta, \Delta, ..., \Delta] \text{ with } N \text{ blocks} \tag{5}$$

The ESRS allows one to manipulate and analyze single-rate and multirate systems as if they were both single-rate. ESRS state space or corresponding transfer function descriptions can be used to calculate input-output relations for systems in series or in feedback loops just as in classical control[14]. For example, to calculate the ESRS of a time-invariant plant in feedback with a multirate compensator, we would calculate the ESRS of the plant and compensator individually, using the same value of T and N for both, and then combine them using block diagram algebra. Furthermore, we could determine the stability of the original closed-loop system by calculating the eigenvalues of the new closed-loop ESRS system[15].

We used the ESRS to evaluate the performance and robustness of our multirate and single-rate flutter suppression systems. First, we formed their closed-loop ESRS's, and then we applied analysis techniques for linear time-invariant systems to the resulting single-rate systems. In the following section, we discuss the results of these analyses.

DESIGN RESULTS

By way of review, our two flutter suppression systems are:

1) Single-Rate Second-Order with TE and LE acceleration inputs and CS command output
2) Multirate Second-Order with multiplexed TE and LE acceleration inputs and CS command output

We compared the performance and robustness of these two compensators in the following areas

1) Gust pulse response
2) Maximum RMS gain from disturbance to the control surface deflection and deflection rate
3) Gain and phase margins at the compensator output
4) The maximum singular value of the minimum destabilizing multiplicative uncertainty at the compensator input

The results are presented for three operating points, mach 0.50 132 psf, mach 0.70 146 psf, and mach 0.78 151 psf. Each of these is 5 psf above the critical flutter dynamic pressure for the corresponding mach number, so the BACT Wing is nominally unstable at each of these operating points. It is important to note that none of these operating points were used for the compensator optimization and so the compensators were not tuned to these particular operating points. Although we will discuss performance and robustness at only three operating points, the following results are indicative of the compensator's performance and robustness at all 24 operating points.

Gust Pulse Response

The gust pulse response provides an indication of the transient response of the closed-loop system to a disturbance input. We computed the gust pulse response by simulating the response of the BACT Wing in feedback with the flutter suppression system to a disturbance input pulse with an amplitude of 10 in/sec and a duration of 0.004 seconds.

Figures 6-7 show the response of the BACT Wing at mach 0.70 and 146 psf to the specified disturbance gust pulse. Also shown is the response of the wing with a continuous LQ regulator. The cost function weights for this LQ regulator design satisfy the same design criterion as were used in the multirate and single-rate designs. The gust pulse responses at the other operating points are similar to those shown in Figures 6-7.

For comparison, gust pulse response plots for the multirate compensator synthesized without fictitious sensor noise are also shown in Figures 6-7. Recall that we added fictitious sensor noise to the multirate design in order to improve the robustness at the compensator input. The primary effect of adding sensor noise is to decrease the damping of the pitch and plunge modes. As can be seen, the reduction in damping is more prevalent in the pitch response than in the plunge response.

Max RMS Gains

One of NASA's specifications was a limit on the steady state covariance of the control surface deflection and deflection rate for a 1 in/sec RMS white noise disturbance. Our closed-loop system consists of a continuous plant and a discrete compensator. Therefore, these steady state covariances are periodically time varying. In Figure 8, we show the steady state covariance propagation for the BACT Wing in feedback with the two compensators at an operating point of mach 0.70 and 146 psf for a unity RMS white noise disturbance.

One meaningful interpretation of NASA's specification would be to look at the peak steady state covariance value taken from the covariance plot. This value, though, is an upper limit on the closed-loop gain for a white noise disturbance and is not a true indicator of the control activity level. A better measure of control activity would be the maximum RMS gain.

The maximum RMS gain of a multirate system is given by

$$\sup_{\mathrm{RMS}(u)\neq 0} \frac{\mathrm{RMS}(y(k))}{\mathrm{RMS}(u(k))} = \sup_{\mathrm{RMS}(u_E)\neq 0} \frac{\mathrm{RMS}(y_E(k))}{\mathrm{RMS}(u_E(k))} = \|G_E(z^N)\|_\infty \qquad (6)$$

where $\|G_E(z^N)\|_\infty$ is the H-infinity norm of the transfer function of the ESRS system between the input and output of interest, u and y respectively. See References 1 and 4 for details.

To apply the discrete equation (6) to our mixed continuous/discrete system, we created a new discrete multirate system in which the continuous inputs and outputs of interest are sampled very fast. We chose a sampling rate for the CS deflection and deflection rate of 500 Hz. This is more than ten times the control surface actuator roll-off frequency. Figure 9 shows a block diagram and sampling schedule of this new discrete time system in feedback with the single-rate compensator. This closed-loop system is *multirate* even though the compensator is *single-rate*. The ESRS for the system has a sample/update rate of 500 Hz and an N of 10.

We used this new ESRS system to estimate the maximum RMS gain of the original single-rate system between the disturbance and the CS deflection and deflection rate. A similar method was used to calculate the maximum RMS gains for the disturbance input to the CS deflection and deflection rate for the multirate flutter suppression system. The results, for the BACT Wing at the three representative operating points, are summarized in Table 2.

Gain and Phase Margins

Gain and phase margins were calculated at the compensator output using the ESRS and a multiloop Nyquist diagram. The ESRS of the plant and compensator were computed independently and then combined in series to form the ESRS loop transfer function. Gain and phase margins were subsequently measured directly off the multiloop Nyquist plot of this function. These are traditional gain and phase margins, and assume that the gain and phase cannot vary simultaneously. The details of this technique are given in References 4 and 16.

The gain and phase margins for the BACT Wing at three operating points are presented in Table 2. These values are typical of the margins at all 24 operating points, although the margins tend to be better at lower dynamic pressures and slightly worse at higher dynamic pressures. A representative Nyquist diagram is shown in Figure 10. This

Table 2. Performance and Robustness Summary

	Mach 0.50, 132 psf		Mach 0.70, 146 psf		Mach 0.78, 151 psf	
	Single-rate	Multirate	Single-rate	Multirate	Single-rate	Multirate
Max RMS Gain Distr. to CS Deflect. (deg sec/in)	0.22	0.25	0.19	0.19	0.11	0.11
Max RMS Gain Distr. to CS Def-rate (deg/in)	6.5	6.9	2.4	2.6	1.5	1.5
Gain Margin	12 db	12 db	10 db	10 db	9 db	9 db
Phase Margin	41°	38°	45°	40°	43°	40°
$\underline{\sigma}\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$	0.41	0.45	0.38	0.44	0.35	0.45
$\underline{\sigma}\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$ w/o sensor noise	0.25	0.35	0.26	0.32	0.25	0.31

particular Nyquist plot has two encirclements of the -1 point because the open-loop plant has two unstable poles.

Robustness at the Compensator Input

The uncertainty at the compensator input was assumed to be a multiplicative perturbation of the form shown in Figure 11, where k_1 and k_2 are complex gains. We transformed this system into the output feedback form traditionally used in robustness analysis using simple block diagram algebra. However, when the compensator is multirate we must use the ESRS of the plant, compensator and uncertainty. A block diagram of this closed-loop ESRS for the multirate flutter suppression system is shown in Figure 12. G_E is the loop transfer function comprised of the compensator and plant ESRS transfer functions connected in series.

Now, given the system in the form shown in Figure 12, we can calculate an exact value for the size of the smallest destabilizing perturbation[17]. First rewrite Δ_E in Figure 12 as

$$\Delta_E = l_1 k_1 + l_2 k_2 \tag{7}$$

where I_1 = diag{1 0 1 0 ... 1 0} with $2N$ diagonal elements, and where I_2 has a similar form. Then it can be shown that[17]

$$\bar{\sigma}(\Delta_{\min}) = \left(\sup_{\phi} \max_{\theta} \rho \left\{ \left[I_1 + I_2 e^{j\theta} \right] H_E(e^{j\phi}) \right\} \right)^{-1} \tag{8}$$

where $\bar{\sigma}(\Delta_{\min})$ represents the maximum magnitude of the smallest destabilizing k_1 or k_2; $0 \leq \phi \leq \pi$; $0 \leq \theta \leq 2\pi$; ρ is the spectral radius; and $H_E(z^N) = G_E(z^N)(I - G_E(z^N))^{-1}$.

We are guaranteed that the system in Figure 11 will remain stable if

$$\bar{\sigma}\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} < \bar{\sigma}(\Delta_{\min}) \tag{9}$$

We are also guaranteed that when (9) is violated, there exist values of k_1 and k_2 that destabilize the system in Figure 11.

Equation (8) is straightforward to solve with a two dimensional search in ϕ and θ. The results are given in Table 2. For comparison, the corresponding results for the design without the fictitious sensor noise are also given in Table 2. Notice that the addition of the fictitious noise increases the maximum singular value of the smallest destabilizing uncertainty by as much as 60%.

Even with the fictitious sensor noise, the robustness at the compensator inputs does not meet NASA's specification of a maximum singular value of 0.75. We could have improved the robustness at the compensator output further by increasing the fictitious sensor noise level, but we chose not to do so because doing so simultaneously reduces the gain and phase margins at the compensator output.

CONCLUSIONS

A new methodology for multirate control system design has been developed. It accommodates a general multiple-input multiple-output control law structure that allows the sampling rates for the plant sensor output signals, the update rates for the processor states, and the update rates for the plant control input signals to be independently specified. It includes a capability to synthesize a single control law for multiple plant conditions so as to achieve robustness to plant parameter variations. Its analysis components include a method for determining conventional gain and phase margins, a method for determining a bound on the smallest destabilizing uncertainty, and a method for determining the maximum RMS gain of a multirate system. As is demonstrated in this paper by application to the design of a multirate flutter suppression system for a model wing, this new methodology is a practical and effective tool for multirate control system design.

REFERENCES

1. Mason, G.S. "Multirate Compensator Synthesis and Analysis," Ph.D. Thesis, University of Washington, Seattle, WA 1992.

2. Mason, G.S., and Berg, M.C., "Reduced-Order Multirate Compensator Synthesis," *AIAA Jour. Guid. Control & Dyn*, Vol 15, No. 3, May-June 1992, pp. 700-706.

3. Berg, M.C., Mason, G.S. and Yang, G.S., "A New Multirate Sampled-Data Control Law Structure and Synthesis Algorithm," *AIAA Jour. Guid. Control & Dyn*, Vol 15, No. 5, Sept.-Oct. 1992, pp. 1183-1191.

4. Mason, G.S. and Berg, M.C., "Robustness Analysis of A Multirate Flutter Suppression System," *AIAA Jour. Guid. Control & Dyn*, Vol. 16, No. 5, Sept-Oct 1993, pp. 992-926.

5. Bennett, R.M., Eckstrom, C.V., Rivera, J.A.Jr., Dansberry, B.E., Farmer, M.G., and Durham, M.H., "The Benchmark Aeroelastic Models Program - Description and Highlights of Initial Results," NASA Tech. Memo. 104180, Dec. 1991.

6. Durham, M.H., Keller, D.F., Bennett, R.M., Wieseman, C.D., "A Status Report on a Model for Benchmark Active Controls Testing," AIAA Paper No. 91-1011, April 1991.

7. Freudenberg, J.S. and Looze, D.P., "Right Half Plane Poles and Zeros and Design Tradeoffs in Feedback Systems," IEEE Trans. Auto. Control, Vol 30, No. 6, June 1985, pp. 555-565.

8. Mukhopadhyay, V., and Newsom, J.R., "A Multiloop System Statility Margin Study Using Matrix Singular Values," *AIAA Jour. Guid., Contr. and Dynamics*, Vol. 7, No. 5, Sept-Oct 1984, pp. 582-587.

9. Meyer, R.A., and Burrus, C.S., "A Unified Analysis of Multirate and Periodically Time-Varying Digital Filters," *IEEE Trans. Circuits in Systems*, Vol. CAS-22, No. 3, March 1975, pp. 162-168.

10. Meyer, D.G., "A New Class of Shift-Varying Operators, Their Shift-Invariant Equivalents, and Multirate Digital Systems," *IEEE Trans. Auto. Control*, Vol. AC-35, No. 4, April 1990, pp 429-433.

11. Ly, U.L., "A Design Algorithm for Robust Low-Order Controllers." Ph.D. Thesis, Stanford Univ., Stanford, CA, 1982.

12. Ly, U.L., "Robust Control Design Using Nonlinear Constrained Optimization," *Proc. Amer. Contr. Conf.*, May 1990, IEEE, Piscataway, NJ, pp. 968-969.

13. Doyle, J.C., and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," *IEEE Trans. Auto. Control*, Vol. AC-26, No. 1, Feb. 1981, pp 4-16.

14. Khargonekar, P.P., Poolla, K. and Tannenbaum, A., "Robust Control of Linear Time-Invariant Plants Using Periodic Compensators," *IEEE Trans. Auto. Control*, Vol. AC-30, No. 11, Nov. 1985, pp. 1088-1096.

15. Kono, M., "Eigenvalue assignment in Linear Periodic Discrete-Time Systems," *Int. J. Control*, Vol. 32, No. 1, 1980, pp. 149-158.

16. Thompson, P.M., "Gain and Phase Margins of Multirate Sampled-Data Feedback Systems," *Inter. Jour. Control*, Vol. 4, No. 3, 1986, pp. 833-846.

17. Doyle, J., "Analysis of Feedback systems with Structured Uncertainties," *IEE Proc.*, Vol 129, Pt. D, No. 6, Nov. 1982, pp. 243-250.

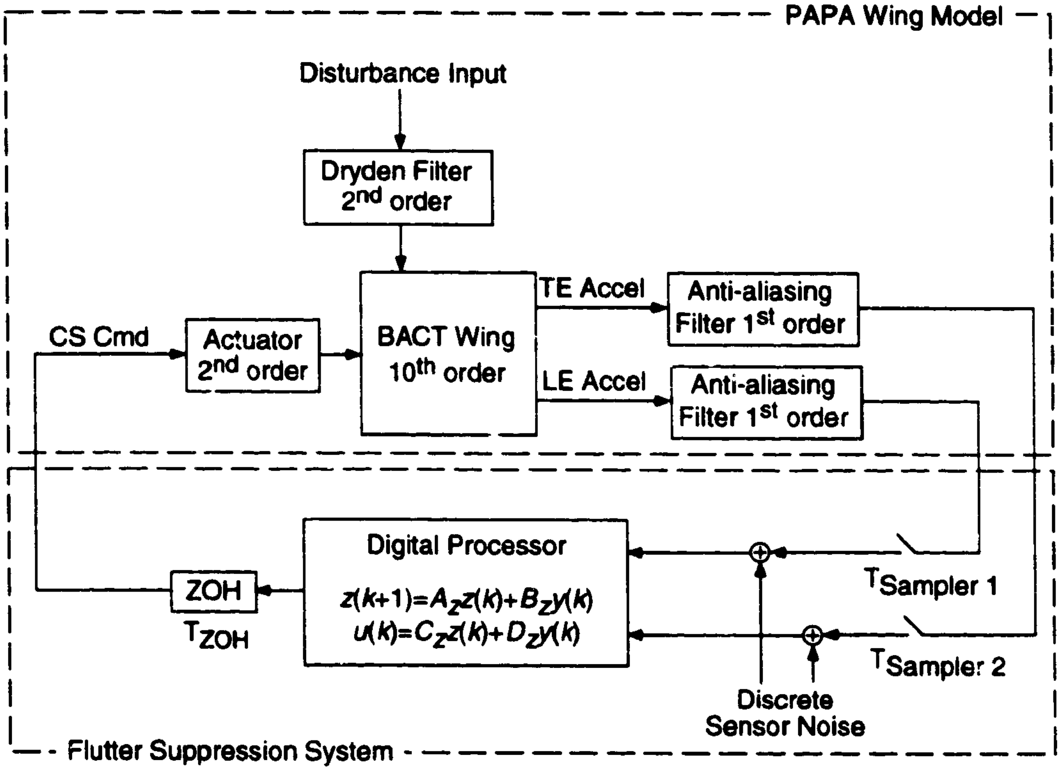


Figure 1. Block diagram of BACT Wing in feedback with a generic flutter suppression system

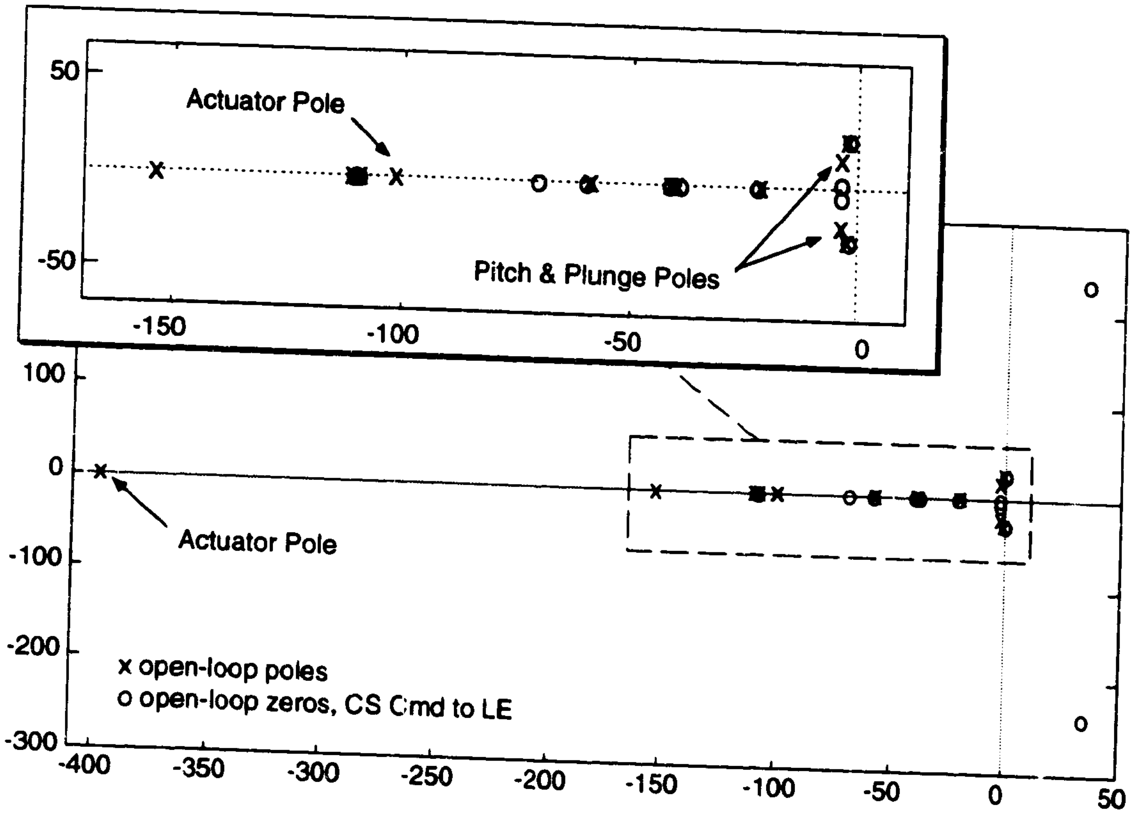


Figure 2a. Open-loop poles and zeros of the CS command to LE accelerometer transfer function at mach 0.50, 75 psf

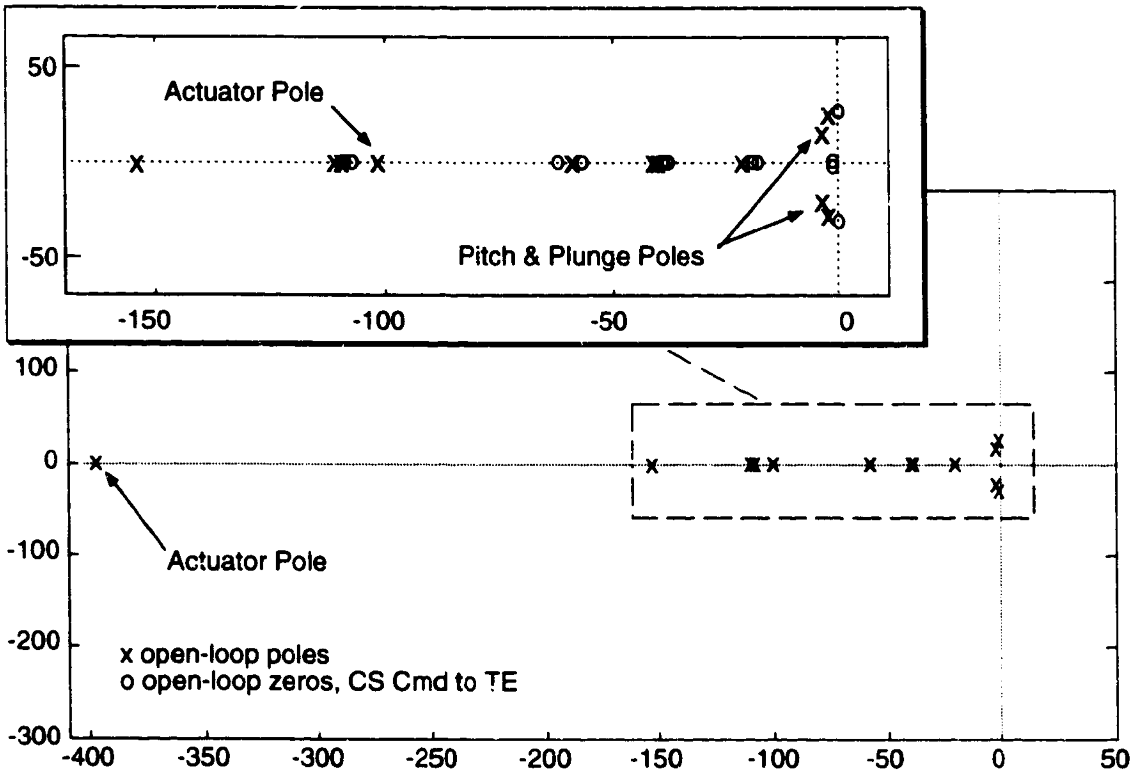


Figure 2b. Open-loop poles and zeros of the CS command to TE accelerometer transfer function at mach 0.50, 75 psf

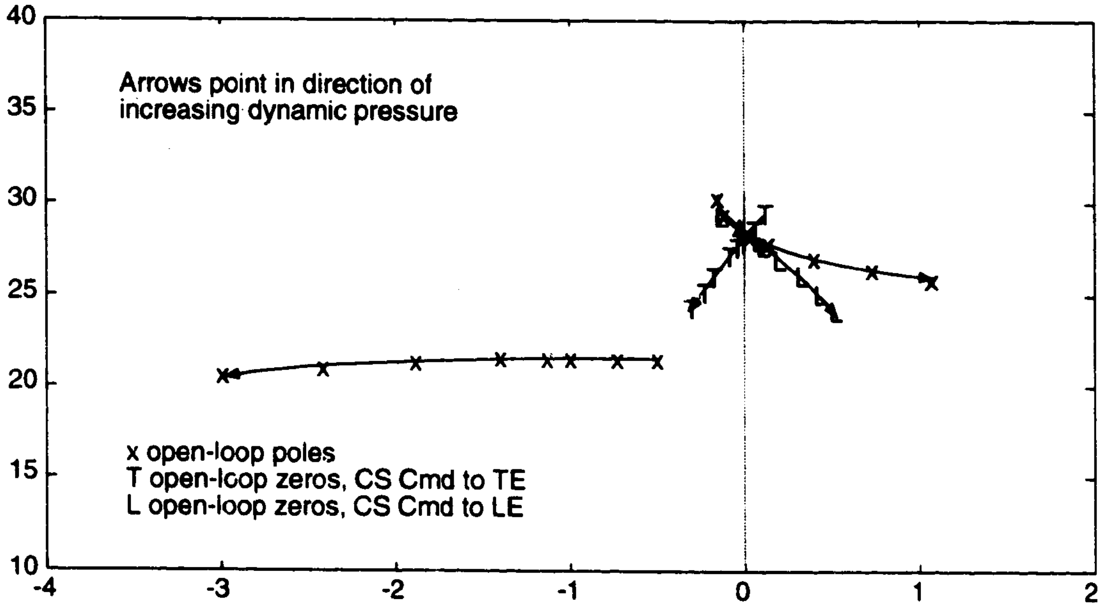


Figure 3. Open-loop poles and zeros versus dynamic pressure at mach 0.5

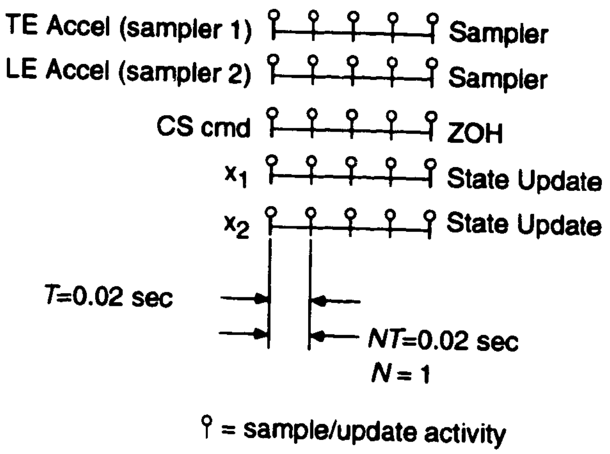


Figure 4. Sampling schedule for single-rate compensator

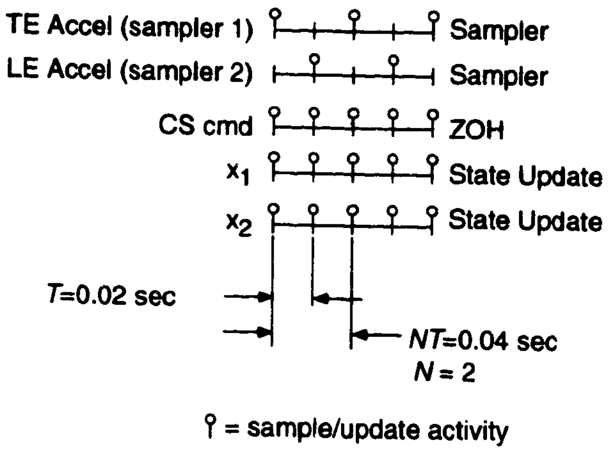


Figure 5. Sampling schedule for multirate compensator

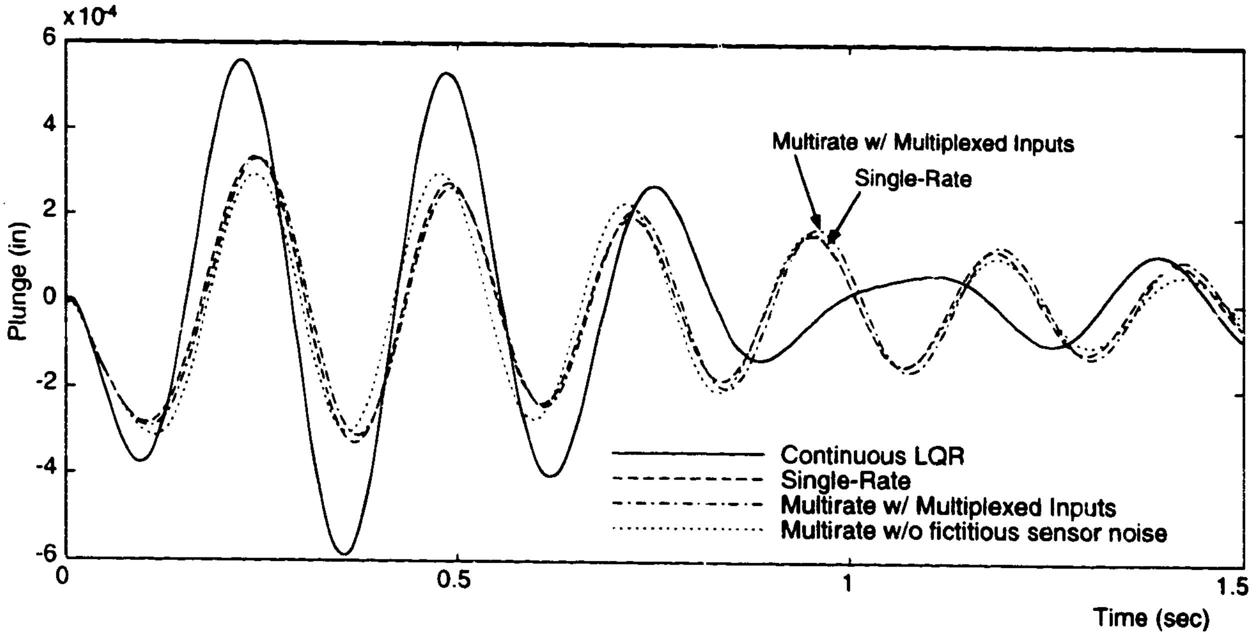


Figure 6. Closed-loop plunge response to gust pulse disturbance at mach 0.70, 146 psf

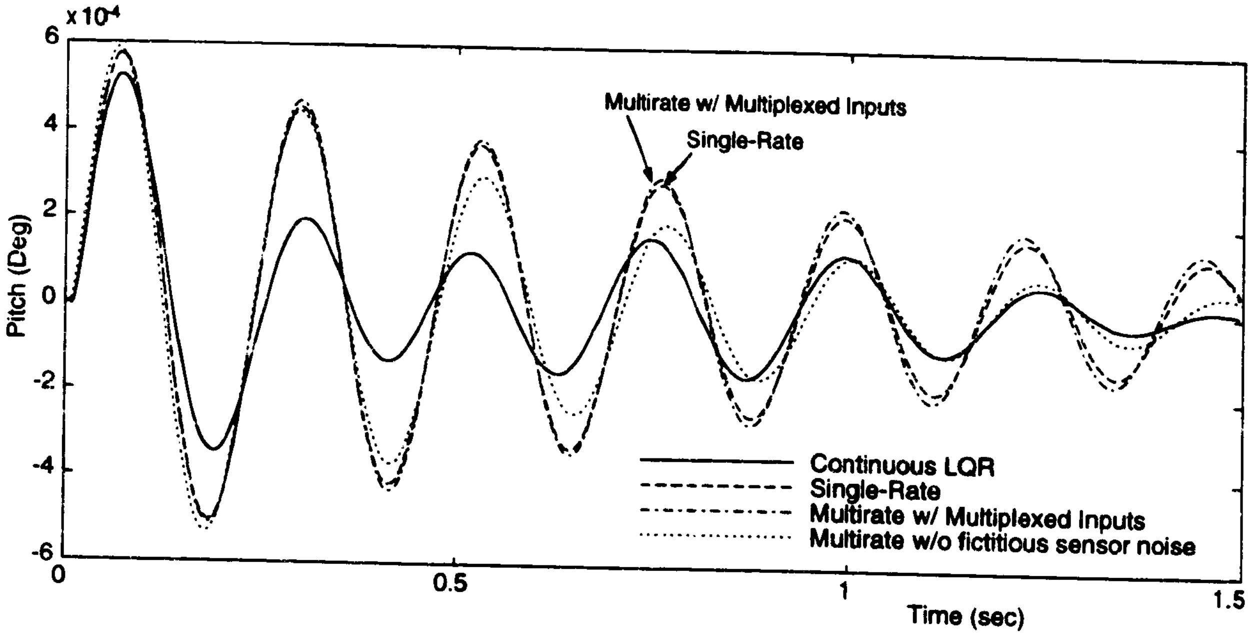


Figure 7. Closed-ioop pitch response to gust pulse disturbance at mach 0.70, 146 psf

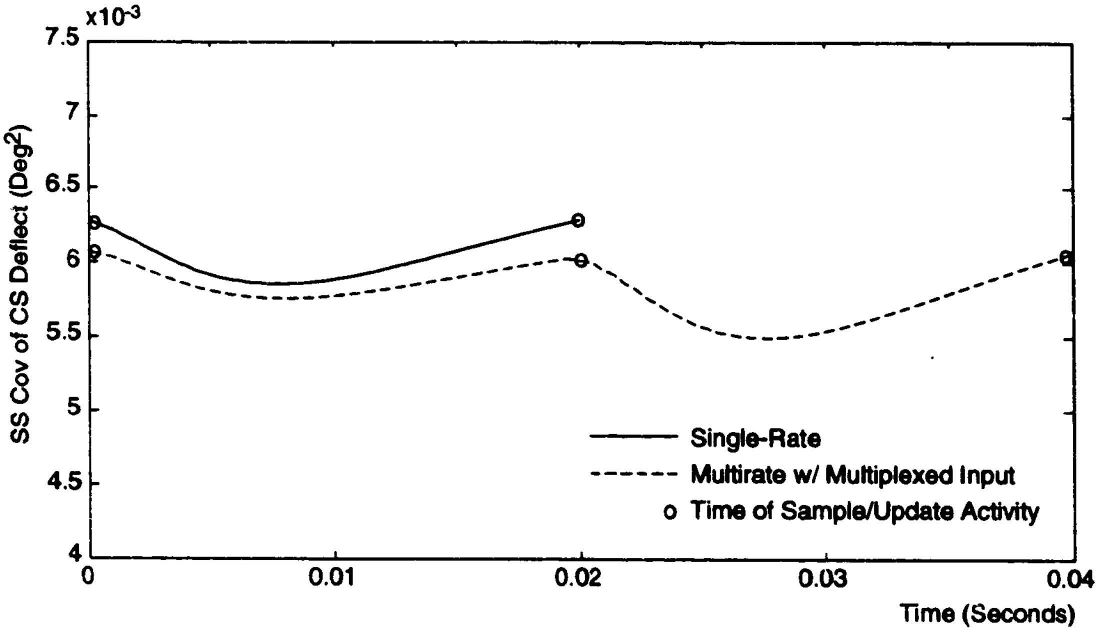


Figure 8. Propagation of steady state covariance of CS deflection at mach 0.70, 146 psf

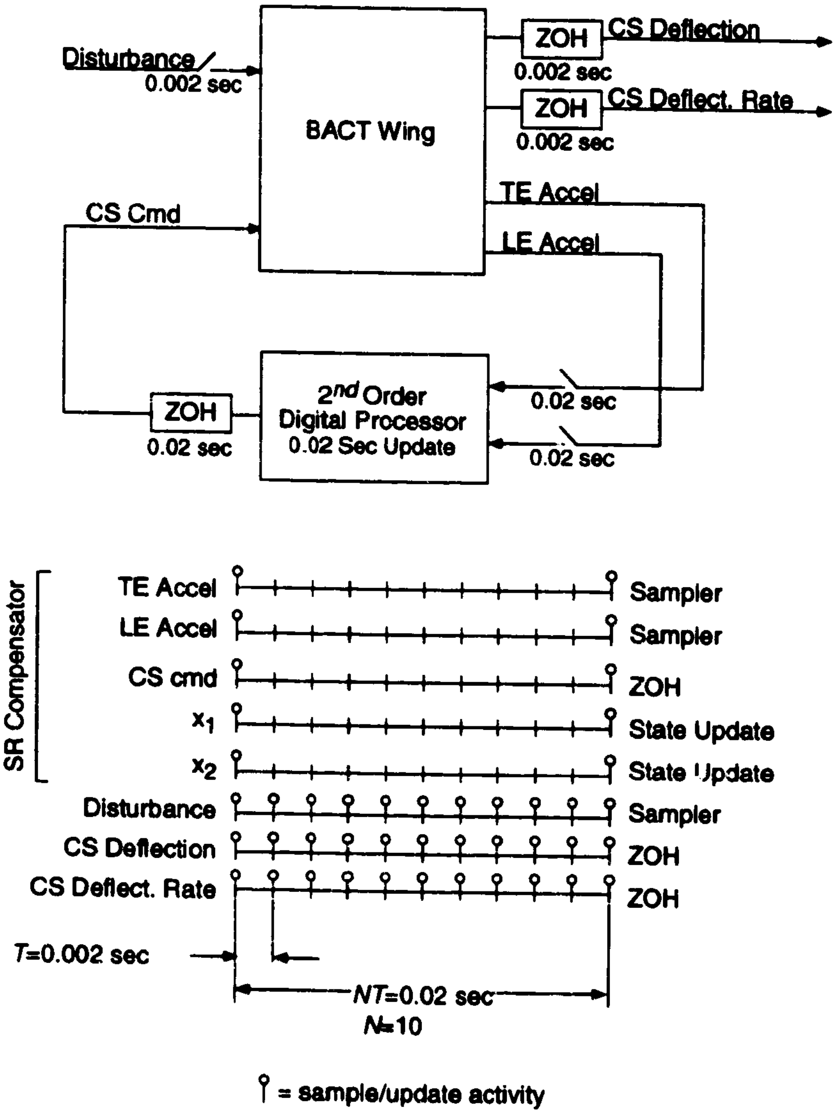


Figure 9. Block diagram and corresponding sampling schedule for the system used to evaluate the maximum RMS gain

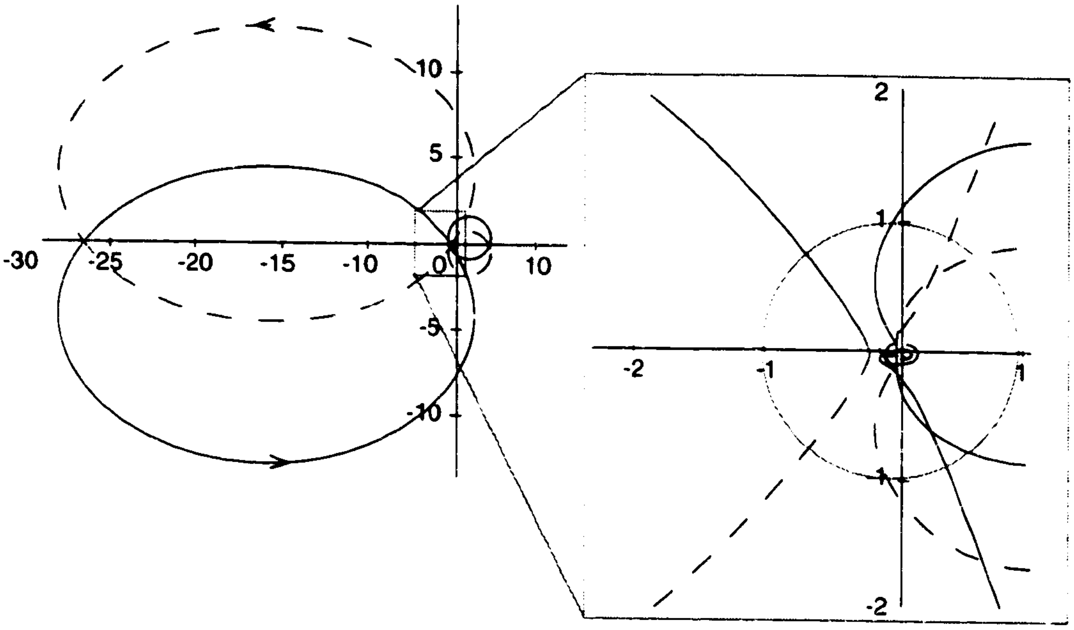


Figure 10. Nyquist diagram of BACT Wing and multirate compensator at mach 0.70, 146 psf with loop broken at the compensator output

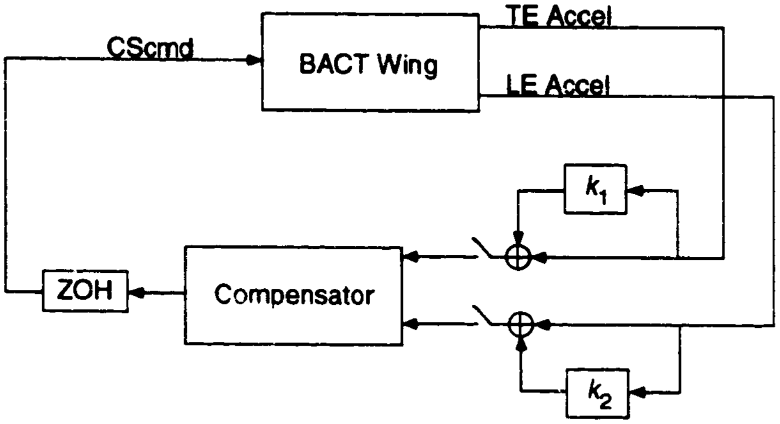
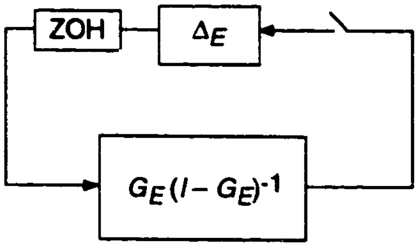


Figure 11. BACT Wing with uncertainty at the compensator input



$$\Delta_E = \text{block diag}\left\{\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}, \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}, \ldots, \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}\right\} \text{ with } N \text{ blocks}$$

Figure 12. ESRS for the system in Figure 11

ATTACHMENT 4

A Collection of Matlab M-Files for

Modeling, Simulation and Synthesis of Multirate Compensators

Department of Mechanical Engineering
University of Washington, 1993

CONTENTS

SECTION ONE

INTRODUCTION

1.0 OVERVIEW

The design and analysis of even a simple multirate compensator can be a complex task. In this document we describe some Matlab M-Files which aid in multirate design. They include M-Files for modeling multirate systems, for computing optimum values of a multirate compensator's gains, and for generating time domain simulations.

The remainder of the document is divided into three sections. In Section Two we review the basics of multirate modeling and our optimization algorithm. We also present the key concepts and notation which are utilized in Section Three. Section Three describes each M-File, detailing its inputs and outputs. The M-Files in this section are divided into three categories: modeling, simulation, and synthesis. Finally, in Section Four we conclude with a multirate design example.

1.1 SOFTWARE REQUIREMENTS

In addition to the standard Matlab toolbox routines, this software uses M-Files from both the Control_Toolbox and the Optimization_Toolbox. These two toolboxes must be present for the multirate software to operate.

SECTION TWO

BACKGROUND

2.0 OVERVIEW

In the following paragraphs we present the notation used in the remaining sections and review some key concepts which will be helpful to those using this software. A detailed explanation of the theory behind our M-Files can be found in References 1-3.

2.1 A MULTIRATE FEEDBACK SYSTEM

A multirate sampled-data system consists of a continuous plant in feedback with a multirate compensator. A block diagram of such a system is shown in Figure 2.1. The vector y_s in this figure represents the continuous plant sensor output. Each element of y_s can be sampled at an independent rate. The vector $\bar{y}$ represents the sampled value of y_s available to the digital processor. (In our double index notation, the discrete signal $p(m,n)$ results from sampling the continuous signal $p(t)$ at the times $t = (mN + n)T$; where the integer N is the period of repetition; T is the sampling time; $m = 0, 1, \ldots$; and $n=0, 1, \ldots, N-1$.) The digital processor obtains the current value of $\bar{y}$ and combines it with the current processor state vector, $\bar{z}$, using the state space structure shown in Figure 2.1. Each element of the processor state vector, $\bar{z}$, can be updated at an independent rate. The continuous output from the compensator, represented by the vector u, is formed by holding the output from the digital processor, $\bar{u}$, with a zero-order-hold. Each element of the vector $\bar{u}$ can be held at an independent rate to form u. The vectors v and w represent the discrete sensor and continuous process noise respectively.

Conceptually, one can divide the multirate compensator into two parts, the "sampling schedule" and the digital processor gains. This is the approach used in the synthesis and analysis software. The "sampling schedule" is a description of when each compensator input is sampled and when each output and processor state is updated, while the digital processor gains determine the dynamics of the digital processor. In the following paragraphs we discuss each in detail.

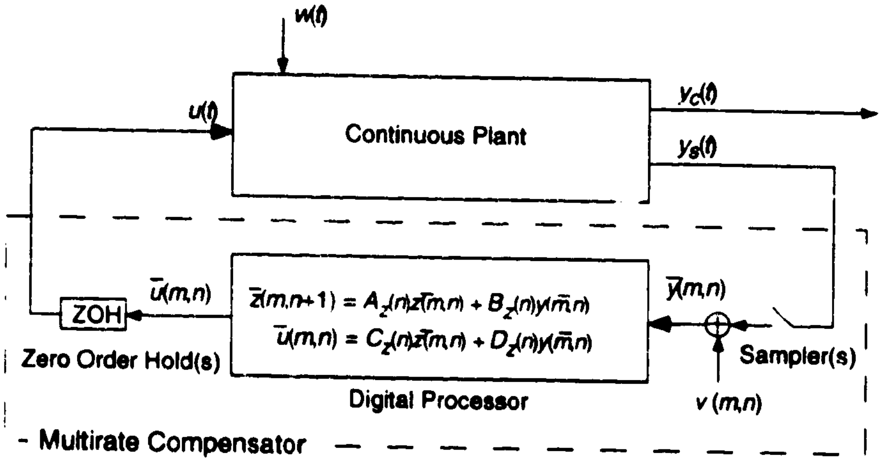


Figure 2.1 Closed-loop Multirate System

2.1.1 The Sampling Schedule

In general, the sampling and updating of y_s, $\bar{z}$, and $\bar{u}$ in Figure 2.1 can occur at any time. We, however, require that these sample and update activities occur only at integer multiples of some fixed time, called the *shortest time period* (STP). The actual value of the STP is arbitrary, but is often a function of the hardware and software used to implement the control law. We also require that the sampling and updating activities of the sensors, states and outputs repeat themselves after some fixed period of time. The period of repetition of the sampling schedule is called the *basic time period* (BTP). Finally, we define

$$\text{the integer } N = \frac{\text{BTP}}{\text{STP}} \quad \text{and the value} \quad T = \text{STP} \tag{2.1}$$

In our double index notation, the first index (m) in $p(m,n)$ indicates the integer number of BTP's which have elapsed when the sample/update occurred and the second index (n) indicates the integer number of STP's which have elapsed within the current BTP when the sample/update occurred.

We can represent the sampling schedule for the multirate compensator graphically, as shown in Figure 2.2. The figure shows a time line for each sampler, processor state, and zero-order-hold. The time line is divided into one STP increments. On the left side of the time line is a description of the signal or state, being sampled or updated. On the right side is a description of the particular activity represented by the time line, e.g. state update, sampler, or zero-order-hold. Circles on each time line indicate when a sample or update activity associated with that particular signal or state takes place. Usually the sampling schedule is shown for only one BTP since the sampling schedule repeats itself every BTP.

In most applications, the sampling/updating activities for a given sensor, output or state will be periodic *within* the BTP, as is shown in Figure 2.2. However, the sampling/updating activities do not have to be periodic within the BTP. The only requirement is that the sampling/updating activities have some period of repetition (the BTP) and that they occur at integer multiples of the STP. Once the STP and BTP have been selected, the designer can arbitrarily specify sampling/updating activities at any multiple of the STP with in one BTP. An example of a multirate sampling schedule in which the sampling/updating activities are not periodic within the BTP is shown in Figure 2.3. A sampling policy like this might be used to multiplex multiple inputs through a single analog to digital converter.

2.1.2 Digital Processor Gains

The processor gains are the values of the matrices A_z, B_z, C_z, and D_z in Figure 2.1. Like the sampling schedule, they can be periodically time-varying with a period of repetition of one BTP. Generally, these matrices are free design parameters which can be adjusted by the designer to improve the performance of the multirate compensator. The synthesis software discussed later in this document can be used to calculate optimum values for these gains.

2.2 THE EQUIVALENT TIME-INVARIANT SYSTEM (ETIS)

A multirate compensator with the structure discussed in Section 2.1 can be modeled as a periodically time-varying single-rate compensator by appending appropriate hold states to the digital processor model. This new compensator has the form

$$x(m,n+1) = A(n)x(m,n) + B(n)u(m,n) \tag{2.2a}$$

$$y_s(m,n) = C(n)x(m,n) + D(n)u(m,n) \tag{2.2b}$$

$$\text{for } n = 0,1,\ldots N\text{-}1, \text{ and } m = 0,1,\ldots$$

This compensator has a sampling period of one STP and a period of repetition of one BTP (or NT). $y_s(m,n)$ represents the values of $y(t)$ sampled every STP; $u(t)$ is formed by holding $u(m,n)$ with a zero-order-hold.

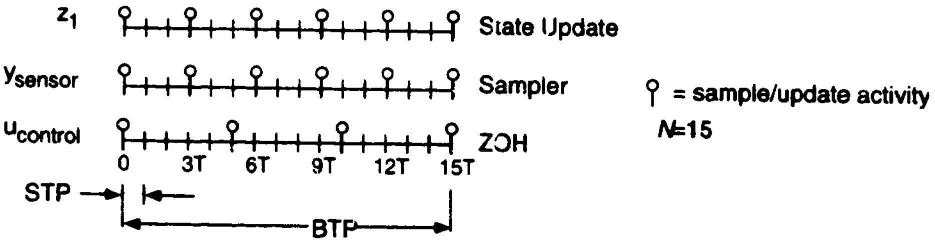


Figure 2.2. Example multirate sampling schedule with periodic sampling/updating activity

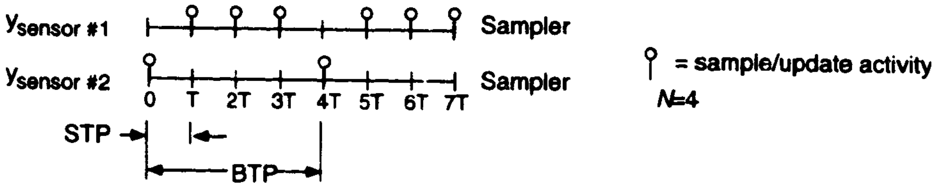


Figure 2.3. Example multirate sampling schedule with aperiodic sampling activity

The periodically time-varying single-rate compensator can be further transformed into a single-rate *Equivalent Time-Invariant System* (ETIS) with the form shown below.

$$x(m+1,0) = A_E x(m,0) + B_E u_E(m,0) \tag{2.3a}$$

$$y_E(m,0) = C_E x(m,0) + D_E u_E(m,0) \tag{2.3b}$$

where

$$\text{and} \quad y_E(m,0) = \begin{bmatrix} y_s(m,0) \\ y_s(m,1) \\ \vdots \\ y_s(m,N\text{-}1) \end{bmatrix} \quad u_E(m,0) = \begin{bmatrix} u(m,0) \\ u(m,1) \\ \vdots \\ u(m,N\text{-}1) \end{bmatrix} \tag{2.4}$$

We use the subscript E to denote vectors and matrices strictly associated with the ETIS. Notice that the ETIS input/output vectors are composite vectors containing the input/output values of the multirate (or periodically-time varying) system at N sampling times. Consequently, an ETIS is always MIMO even if the original system is SISO. If the multirate system has p inputs, q outputs and a sampling period of one STP then the ETIS is a single-rate linear *time-invariant* system with Np inputs, Nq outputs and a sampling period of one BTP.

The ETIS is fundamental to the analysis of multirate systems. It allows one to evaluate the performance and stability of complex systems comprised of multirate, periodically time-varying and/or single-rate components using only modified linear time-invariant single-rate techniques. For example, to evaluate the stability of the system in Figure 2.1, we would first transform the multirate compensator into its ETIS with a given value for N. Then we would discretize the plant at the STP of the compensator using a zero-order-hold and transform the resulting single-rate system into an ETIS using the BTP of the compensator. Next, the plant and compensator ETIS's could be combined in feedback just as if they were traditional single-rate systems. Finally, we could determine the stability of the original multirate sampled-data system from the eigenvalues of its closed-loop ETIS.

2.3 OPTIMIZING THE DIGITAL PROCESSOR GAINS

The synthesis software discussed later calculates optimum values of the digital processor gains, A_z, B_z, C_z, and D_z, by minimizing a quadratic cost function which reflects the peiiormance of the closed-loop system in Figure 2.1. The cost function has the form

$$J = \lim_{t\to\infty} E\left\{\begin{bmatrix} y_c^T(t) & u^T(t)\end{bmatrix}\begin{bmatrix} Q_1 & M \\ M^T & Q_2\end{bmatrix}\begin{bmatrix} y_c(t) \\ u(t)\end{bmatrix}\right\} \tag{2.5}$$

where J is the cost associated with the closed-loop system shown in Figure 2.1. The vector y_c is the continuous criterion output and u is the continuous control input Q_1, Q_2 and M are the cost function weighting matrices which are selected by the designer.

This cost function has the same form as that minimized by a continuous time LQR design. Thus the cost associated with the optimized multirate compensator and that of an LQR design can be compared directly. The designer can also use this fact to help select appropriate values for Q_1, Q_2 and M.

To improve the robustness of the compensator, the optimization algorithm can optimize the digital processor gains for several different plant perturbations simultaneously. The resulting compensator will stabilize the plant at each perturbation and provide overall optimum performance. This is accomplished by minimizing the new cost function of Eqn. (2.6) which is the sum of the costs associated with each plant perturbation.

$$J = \sum_{i=1}^{\mathbf{Np}} J_i = \sum_{i=1}^{\mathbf{Np}} \lim_{t\to\infty} E\left\{\begin{bmatrix} y_{ci}^T(t) & u_i^T(t)\end{bmatrix}\begin{bmatrix} Q_i & M_i \\ M_i^T & R_i\end{bmatrix}\begin{bmatrix} y_{ci}(t) \\ u_i(t)\end{bmatrix}\right\} \tag{2.6}$$

Here J_i is the cost associated with the i^{th} plant perturbation and there are **Np** plant perturbations.

Optimum values of A_z, B_z, C_z, and D_z, occur when

$$\frac{\partial J}{\partial A_z} = 0,\ \frac{\partial J}{\partial B_z} = 0,\ \frac{\partial J}{\partial C_z} = 0,\ \text{and}\ \frac{\partial J}{\partial D_z} = 0 \tag{2.7}$$

Our algorithm uses a gradient type numerical search to determine values of the digital processor gains such that the conditions in Eqn. (2.7) are satisfied. Because the synthesis software uses an iterative process to detennine optimum values for the digital processor gains, the user must provide the software with an initial guess for A_z, B_z, C_z, and D_z. The compensator corresponding to these values must stabilize every plant perturbation considered in (2.6). Obtaining a suitable stabilizing initial guess can sometimes be a difficult problem. We refer the interested reader to Reference 1.

SECTION THREE

M-FILE DOCUMENTATION

3.0 STANDARD VARIABLE DEFINITIONS

Many of the M-Files discussed in this document require similar input variables or provide similar outputs. To simplify the documentation of these M-Files a set of standard variables are used throughout this document. They are defined in Table 3.1 with Matlab variables and functions bolded.

Table 3.1 Standard Variable Definitions

Variable — Description

plt Plant matrices in the form **plt** = $[F, G; H, J]$ where the state-space representation of the plant is

$$\dot{x}(t) = Fx(t) + Gu(t) \tag{3.1a}$$

$$y(t) = Hx(t) + Ju(t) \tag{3.1b}$$

or

$$x(m,n+1) = Fx(m,n) + Gu(m,n) \tag{3.1c}$$

$$y(m,n) = Hx(m,n) + Ju(m,n) \tag{3.1d}$$

depending on the value of **stp** defined later.

nplt Number of states in (3.1), or equivalently the number of rows in F

cmp Multirate compensator gain matrices. Given the state-space representation of the digital processor

$$\bar{z}(m,n+1) = A_z(n)\bar{z}(m,n) + B_z(n)\bar{y}(m,n) \tag{3.2a}$$

$$\bar{u}(m,n) = C_z(n)\bar{z}(m,n) + D_z(n)\bar{y}(m,n) \tag{3.2b}$$

where m=0,1,... and n=0,1,...,N-1, so that the gain matrices, A_z, B_z, C_z and D_z, are periodically time-varying, **cmp** has the form

$$\mathbf{cmp} = \left[\begin{bmatrix} A_z(0) & B_z(0) \\ C_z(0) & D_z(0) \end{bmatrix}, \begin{bmatrix} A_z(1) & B_z(1) \\ C_z(1) & D_z(1) \end{bmatrix}, \ldots, \begin{bmatrix} A_z(N-1) & B_z(N-1) \\ C_z(N-1) & D_z(N-1) \end{bmatrix}\right] \tag{3.3}$$

If A_z, B_z, C_z, and D_z are constant, then

$$\mathbf{cmp} = \begin{bmatrix} A_z(0) & B_z(0) \\ C_z(0) & D_z(0) \end{bmatrix} \tag{3.4}$$

and it is assumed that $A_z(0)=A_z(1)= \ldots =A_z(N-1)$, $B_z(0)=B_z(1)= \ldots =B_z(N-1)$, etc.

The software automatically deduces from the size of **cmp**, the value of **ncmp** and the number of compensator inputs whether the digital processor gains are periodically time-varying or not.

Continued on following page...

Table 3.1 Standard Variable Definitions (*continued from previous page*)

Variable — **Description**

ncmp — Number of digital processor states in Eqn. (3.2), or equivalently the number of rows in A_z.

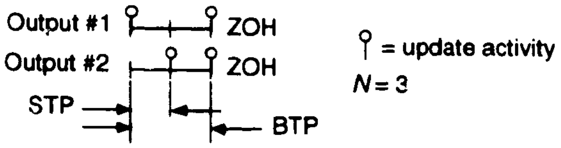
stp — If **stp** > 0 then **stp** is STP, the shortest sample/update period of the compensator. See Section 2.1.1, and **plt** describes a continuous plant.

If **stp** = **0** then the plant matrices **plt** describe a discrete system whose sampling rate is one STP of the compensator, and **plt** describes a discrete plant.

stppbtp — Number of STP's per BTP. **stppbtp** = N. See Section 2.1.1.

su — Sampling schedule for the compens 'or output. **su** has two forms

Case I: **su** is a matrix. **su** must have as many columns as there are compensator outputs, and must have N rows (N = BTP/STP). Each element of **su** has a value of 1 or 0. A 1 in the i^{th} column and j^{th} row of **su** indicates that the i^{th} compensator output is updated at the j^{th} STP within the current BTP. A 0 indicates no update and the previous value is held. For example, given a two output system with the following sampling schedule



the corresponding value of **su** is

$$\mathbf{su} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \tag{3.5}$$

Case II: **su** is a row vector. The i^{th} column of **su** specifies the sampling period of the i^{th} compensator output hold in number of STP's. For example, **su**=[1 4] specifies that the first output is updated every STP and the second output is updated every fourth STP. This form assumes that the updating is synchronized on the first STP of the BTP.

sy — Sampling schedule for the compensator input. **sy** has the same form as **su** except that it specifies when the plant output (the compensator input) is sampled.

sz — Sampling schedule for the compensator states, **sz** has the same form as **su** except that it specifies when the compensator's digital processor states are updated.

3.1 MODELING M-FILES

3.1.1 mr2etis

Format: **[ae,be,ce,de]=mr2etis(cmp, ncmp, su, sy, sz, stppbtp)**

Description: Converts a multirate system into an ETIS with the form:

$$x(m+1,0)=\mathbf{ae}x(m,0)+\mathbf{be}\begin{Bmatrix} y(m,0)\\ y(m,n+1)\\ \vdots\\ y(m,N-1)\end{Bmatrix} \tag{3.6a}$$

$$\begin{Bmatrix} u(m,0)\\ u(m,n+1)\\ \vdots\\ u(m,N-1)\end{Bmatrix}=\mathbf{ce}x(m,0)+\mathbf{de}\begin{Bmatrix} y(m,0)\\ y(m,n+1)\\ \vdots\\ y(m,N-1)\end{Bmatrix} \tag{3.6b}$$

Inputs: **cmp, ncmp, su, sy, sz, stppbtp** The compensator description. See Table 3.1.

Outputs: **ae, be, ce, de** The ETIS matrices in Eqn. (3.6)

3.1.2 mr2ptv

Format: **[P, nu, ny, nz]=mr2ptv(cmp, ncmp, su, sy, sz, stppbtp)**

Description: Converts a multirate system into a periodically time-varying system of the form

$$x(m,n+1)=A(n)x(m,n)+B(n)y(m,n) \tag{3.7a}$$

$$u(m,n)=C(n)x(m,n)+D(n)y(m,n) \tag{3.7a}$$

Inputs: **cmp, ncmp, su, sy, sz, stppbtp** The compensator description. See Table 3.1.

Outputs: **P** The periodically time-varying system matrices in Eqn. (3.7), where

$$\mathbf{P}=\left[\begin{bmatrix} D(0) & C(0)\\ B(0) & A(0)\end{bmatrix},\begin{bmatrix} D(1) & C(1)\\ B(1) & A(1)\end{bmatrix},\ldots,\begin{bmatrix} D(N-1) & C(N-1)\\ B(N-1) & A(N-1)\end{bmatrix}\right] \tag{3.8}$$

nu Number of inputs u in Eqn. (3.7)

ny Number of outputs y in Eqn. (3.7)

nz Number of states x in Eqn. (3.7)

3.1.3 ptv2etis

Format: **[ae, be, ce, de]=ptv2etis(P, nu, ny, nz, stppbtp)**

Description: Converts a periodically time-varying system into its ETIS of the form of Eqn. (3.6)

Inputs: **P, nu, ny, nz** The periodically time-varying system matrices. See Section 3.1.2
stppbtp See Table 3.1

Outputs: **ae, be, ce, de** The ETIS matrices. See Eqn. (3.6)

3.1.4 sr2etis

Format: **[ae, be, ce, de]=sr2etis(a, b, c, d, stppbtp)**

Description: Converts the single-rate system in Eqn. (3.9) into an ETIS with an N of **stppbtp**, and the form of Eqn. (3.6)

$$x(m,n+1) = a(n)x(m,n) + b(n)y(m,n) \tag{3.9a}$$

$$u(m,n) = c(n)x(m,n) + d(n)y(m,n) \tag{3.9b}$$

Inputs: **a, b, c, d** The single-rate systems matrices. See Eqn. (3.9)
stppbtp The desired N of the new ETIS. See Table 3.1

Outputs: **ae, be, ce, de** The ETIS matrices. See Eqn. (3.6)

3.1.5 stack

Format: **[y]=stack(w, stppbtp)**

Description: Converts a traditional discrete-time vector **w** into an ETIS vector **y**

Inputs: **w** A matrix whose columns contain the values of the vector $y(m,n)$ for Nk successive samples times, where k is an integer. The matrix **w** has the form

$$\mathbf{w} = [\,y(0,0),\ y(0,1),\ \ldots\ ,y(0,N\text{-}1)\ y(1,0),\ y(1,1),\ \ldots,\ y(1,N\text{-}1),\ \ldots,\ y(k\text{-}1,N\text{-}1)] \tag{3.10}$$

y is written using the double index notation of Section 2.1.
stppbtp See Table 3.1.

Outputs: **y** An ETIS vector of the values of y_E (the ETIS of y) with the form

$$\mathbf{y} = [\,y_E(0,0), y_E(1,0), \ldots, y_E(k-1,0)] = \left[\begin{bmatrix} y(0,0) \\ y(0,1) \\ \vdots \\ y(0,N-1) \end{bmatrix} \begin{bmatrix} y(1,0) \\ y(1,1) \\ \vdots \\ y(1,N-1) \end{bmatrix} \cdots \begin{bmatrix} y(k-1,0) \\ y(k-1,1) \\ \vdots \\ y(k-1,N-1) \end{bmatrix}\right] \tag{3.11}$$

3.1.6 unstack

Format: [w]=**unstack(y, stppbtp)**

Description: Converts an ETIS vector **y** into a traditional discrete-time vector **w**.

Inputs: **y** An ETIS vector of the form of Eqn. (3.11)
stppbtp See Table 3.1

Outputs: **w** A vector of the values of $y(m,n)$ as in Eqn. (3.10)

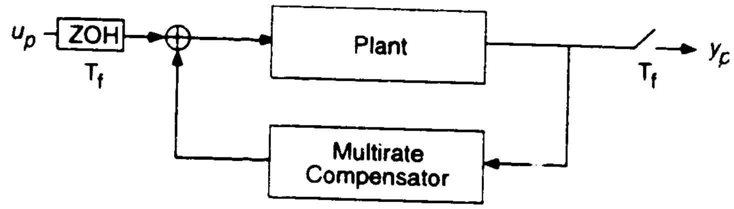
NOTE: Additional M-Files used in support of those described in this section are described in Appendix B.

3.2 SIMULATION M-FILES

3.2.1 mrsim

Format: [y, t] =**mrsim(plt, nplt, cmp, ncmp, su, sy, sz, stppbtp, simpstp, inputl, outputl, ur, stp, X0)**

Description: Generates a time response simulation of a continuous or discrete plant in feedback with a multirate compensator. The routine assumes positive feedback as shown below with the continuous inputs and outputs of interest sampled or updated every **stp/simpstp** seconds. This configuration allows the user to simulate the inter-sample response of the multirate system.



T_f = **stp/simpstp**

Inputs:

plt, nplt A description of the plant, see Table 3.1.

cmp, ncmp, su, sy, sz, stppbtp,A description of the multirate compensator, see Table 3.1.

simpstp This routine can calculate the response of the closed-loop system between the sample/update activities when the plant is continuous (**stp** > 0). **simpstp** is the number of simulation steps per STP of the compensator. **simpstp** must be an integer greater than zero and **simpstp** = 1 if **stp** = 0.

inputl A vector specifying which plant inputs are connected to the compensator outputs. For example **inputl** = [1 4] indicates that the compensator outputs are connected to the first and fourth plant inputs.

outputl A vector specifying which plant outputs are connected to the compensator inputs. For example **outputl** = [2 4] indicates that the compensator inputs are connected to the second and fourth plant outputs.

ur The discrete input vector containing the values of the input at every (**stp/simpstp**) time for $k\cdot$**stppbtp**$\cdot$**simpstp** sample times. **ur** has the form

$$\mathbf{ur} = [\, u_p(0,0),\ u_p(0,1),\ \ldots\ , u_p(k\text{-}1, \mathbf{stppbtp}\cdot\mathbf{simpstp} - 1)] \qquad (3.12)$$

Note that in Eqn. (3.12) the sample/update period is one **stp/simpstp**. This sample period should not be confused with the sample period of the multirate compensator which is simply one **stp.** The second index in the double index notation in Eqn. (3.12) takes on values between zero ard **simpstp**$\cdot$**stppbtp**.

stp see Table 3.1.

X0 A vector containing the initial conditions of the plant state vector. If **X0** is omitted, the initial conditions are assumed to be zero.

Outputs:

y A vector containing the response of the plant every (**stp/simpstp**) time. **y** has the form

$$\mathbf{y} = [\, y_p(0,0),\ y_p(0,1),\ \ldots\ , y_p(k\text{-}1, \mathbf{stppbtp}\cdot\mathbf{simpstp} - 1)] \qquad (3.13)$$

Note that the second index in the double index notation corresponds to the second index of Eqn. (3.12)

t a vector containing the times corresponding to the columns of **y**. When **stp** = 0 then **t** indicates the number of sampling periods.

Comments: This M-File uses the ETIS of the system to compute the time domain response. Hence, if k in Eqn. (3.12) is not a multiple of **simpstp**$\cdot N$ where (N=BTP/STP) the system response is only simulated to the nearest multiple of **simpstp**$\cdot N$

3.2.2 mrfeedback

Format: **[ae, be, ce, de] =mrfeedback(plt, nplt, cmp, ncmp, su, sy, sz, stppbtp, inputl, outputl, stp)**

Description: Creates a closed-loop ETIS system from a discrete or continuous plant, depending on **stp**, and a multirate compensator using *positive* feedback.

Inputs: **plt, nplt** The plant description. See Table 3.1.

cmp, ncmp, su, sy, sz, stppbtp The multirate compensator description. See Table 3.1.

inputl A vector specifying which plant inputs are connected to the compensator outputs. For example **inputl** = [1 4] indicates that the compensator outputs are connected to the first and fourth plant inputs.

outputl A vector specifying which plant outputs are connected to the compensator inputs. For example **outputl** = [2 4] indicates that the compensator inputs are connected to the second and fourth plant outputs.

stp See Table 3.1. If **stp** is omitted **mrfeedback** assumes **stp**=0.

Outputs: **ae, be, ce, de** The system matrices of the ETIS closed-loop system with the form of Eqn. (3.6). The inputs and outputs of this system are the inputs and outputs of the discrete plant

3.3 SYNTHESIS M-FILES

The synthesis algorithm is implemented as a series of script files. They allow the designer to optimize the performance of the closed-loop system shown in Figure 3.1 by calculating values of A_z, B_z, C_z, and D_z, such that the cost function of Eqn. (2.6) is minimized. There are three distinct steps to the optim·zation process: 1) entering the data which describes the problem; 2) discretizing and preprocessing the data; and 3) optimizing the compensator gain values. Each step of the optimization process is described in detail in the following paragraphs.

Although the synthesis algorithm was designed to solve the multirate sampled-data problem, it is fairly general and can be applied to a variety of related problems as well. It can compute optimum compensator gains for a sampled-data system consisting of a continuous plant in feedback with a discrete compensator as shown in Figure 2.1, or for a discrete system consisting of a discrete plant in feedback with a discrete compensator. In either case, the compensators may be single-rate or multirate and may have either time-invariant or periodically time-varying digital processor gains.

3.3.1 Input Variables

The synthesis algorithm looks for specific Matlab "workspace" variables to define the optimization problem. (A "workspace" variable is defined from the » prompt as opposed to being passed as an argument to a function.) These variables define things such as the plant dynamics, the cost function and the compensator structure and are defined in Table 3.2. The user must assign values to these variables before beginning the optimization. This can either be done manually or automatically from a script. An example of a script which defines these variables is provided in Appendix D.

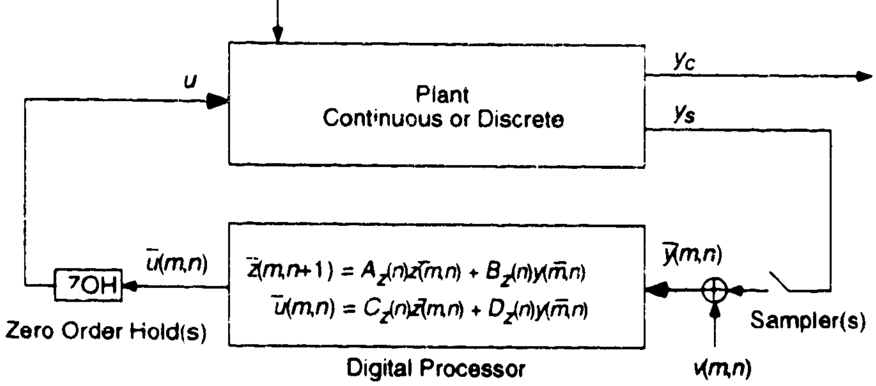


Figure 3.1 Closed-loop Multirate System

Table 3.2 Matlab Workspace Variables Defining Optimization Problem

Variable — Description

Np

The number of plant perturbations. **Np**=1 indicates only the nominal pant is defined. See Section 2.3

plt

State-space matrices of the plant(s). Given the i^{th} continuous plant perturbation

$$\dot{x}_i(t) = F_i x_i(t) + G_i \begin{Bmatrix} u_i(t) \\ w_i(t) \end{Bmatrix} \tag{3.14a}$$

$$\begin{Bmatrix} y_{ci}(t) \\ y_{si}(t) \end{Bmatrix} = H_i x_i(t) + J_i \begin{Bmatrix} u_i(t) \\ 0 \end{Bmatrix} \tag{3.14b}$$

or the i^{th} discrete plant perturbation, depending on the value of **stp** defined later,

$$x_i(m,n+1) = F_i x_i(m,n) + G_i u_i(m,n) + w_i(m,n) \tag{3.14c}$$

$$\begin{Bmatrix} y_{ci}(m,n) \\ y_{si}(m,n) \end{Bmatrix} = H_i x_i(m,n) + J_i \begin{Bmatrix} u_i(m,n) \\ 0 \end{Bmatrix} \tag{3.14d}$$

where u is the control input; w_i is zero mean white noise input.; y_{ci} is the criterion output; and y_{si} are the outputs which are sampled by the compensator. Then

$$\mathbf{plt} = \left[\begin{bmatrix} F_1 & G_1 \\ H_1 & J_1 \end{bmatrix}, \begin{bmatrix} F_2 & G_2 \\ H_2 & J_2 \end{bmatrix}, \ldots, \begin{bmatrix} F_{\mathbf{Np}} & G_{\mathbf{Np}} \\ H_{\mathbf{Np}} & J_{\mathbf{Np}} \end{bmatrix} \right] \tag{3.15}$$

For convenience we assumed in Eqns. (3.14) that the input vector is partitioned into the control input u_i and the disturbance input w_i. The software, however, can accommodate a more general form for the input vector. Using the variables **ucol** and **ncol**, defined later, the user can specify which inputs correspond to control and noise inputs respectively. Any row of the input vector can be either control, disturbance or both. (We interpret an input vector specified as both control and disturbance as two separate inputs, one a control input and the other a disturbance input whose distribution vectors have the same values.)

Similarly, the outputs y_{ci} and y_{si} are specified by **crow** and **srow** respectively, and any row of the output vector can be criterion output, sensor output or both.

NOTE: For a discrete plant, the process noise distribution matrix is unity.

wxx

A vector of the PSD values of the white noise disturbance w_i. If

$$W_i \delta(\tau) = E\{w_i(t-\tau) w_i^T(t)\} \tag{3.16a}$$

or if

$$W_i(\delta(k) + \delta(j)) = E\{w_i(m,n) w_i^T(m+k,n+j)\} \tag{3.16b}$$

depending on the value of **stp**, then

$$\mathbf{wxx} = [W_1 \;\; W_2 \; \ldots \; W_{\mathbf{Np}}]$$

continued on following page...

Table 3.2 Matlab Workspace Variables Defining Optimization Problem *(continued from previous page)*

Variable — Description

vr The PSD values of the discrete sensor noise covariance. If the output, sampled every STP, is

$$y_{si}(k) = H_i x_i(k) + v_i(k) \qquad (3.17)$$

where $V_i \delta(j) = E\{v_i(k) v_i^T(k+j)\}$

then

$$\mathbf{vr} = [\,V_1 \; V_2 \; \ldots \; V_{\mathbf{Np}}\,]$$

ncol A vector specifying which columns of G_i correspond to w_i. If **ncol** = [1 4], then the plant disturbance vector w is comprised of the first and fourth plant inputs.

ucol A vector specifying which columns of G_i correspond to u_i. If **ucol** = [2 4], then the plant control input vector u is comprised of the second and fourth plant inputs. (Again, we interpret an input vector specified as both control and disturbance as two separate inputs, one a control input and the other a disturbance input whose distribution vectors have the same values.)

srow A vector specifying which rows of C_i correspond to y_{si}. If **srow** = [1 3], then the first and third continuous plant outputs are sampled and connected to the compensator.

crow A vector specifying which rows of C_i correspond to y_{ci}. If **srow** = [1 3], then the first and third continuous plant outputs are used to calculate the cost given in Eqn. (3.18).

q1a, **q2a**, **ma**The cost function weighting matrices for either a continuous or discrete cost function, depending on the value of **stp** defined later. The cost function has the form

$$J_{continuous} = \lim_{t \to \infty} \sum_{i=1}^{\mathbf{Np}} E\left\{ \begin{bmatrix} y_{ci}^T(t) & u_i^T(t) \end{bmatrix} \begin{bmatrix} Q_{1i} & M_i \\ M_i^T & Q_{2i} \end{bmatrix} \begin{bmatrix} y_{ci}(t) \\ u_i(t) \end{bmatrix} \right\} \qquad (3.18a)$$

or

$$J_{discrete} = \lim_{m \to \infty} \frac{1}{\mathbf{Stppbtp}} \sum_{i=1}^{\mathbf{Np}} \sum_{n=0}^{\mathbf{Stppbtp}-1} E\left\{ \begin{bmatrix} y_{ci}(m,n) \\ u_i(m,n) \end{bmatrix}^T \begin{bmatrix} Q_{1i} & M_i \\ M_i^T & Q_{2i} \end{bmatrix} \begin{bmatrix} y_{ci}(m,n) \\ u_i(m,n) \end{bmatrix} \right\} \qquad (3.18b)$$

where i corresponds to a particular plant condition.

$$\mathbf{q1a} = [\,Q_{11} \; Q_{12} \ldots Q_{1\mathbf{Np}}]$$
$$\mathbf{q2a} = [\,Q_{21} \; Q_{22} \ldots Q_{2\mathbf{Np}}]$$
$$\mathbf{ma} = [\,N_1 \; N_2 \ldots N_{\mathbf{Np}}\,]$$

qa and **ra** must be symmetric and positive semi-definite.

If **ma** is left undefined its value is assumed to be zero.

continued on following page.

Table 3.2 Matlab Workspace Variables Defining Optimization Problem *(continued from previous page)*

Variable	Description
cmp	The compensator gain matrices as defined in Table 3.1. The digital processor gains can be either periodically time-varying or time-invariant. IMPORTANT: The user must provide an initial guess for **cmp** which stabilizes *all* plant perturbations.
cmpfree	This matrix specifies which of the gains in **cmp** are fixed. **cmpfree** has the same dimensions as **cmp**, but its elements are either 0 or 1. A 0 indicates that the corresponding element in **cmp** is fixed and should not be optimized. A 1 indicates that the corresponding element in **cmp** is free and can be optimized.
su, sy, sz	The sampling schedule description. See Table 3.1. By selecting appropriate values for **su, sy, sz** and **Stppbtp** the designer can specify whether the compensator is multirate or single-rate.
Stppbtp	The same as **stppbtp** in Table 3.1. Note that the first letter of **Stppbtp** is upper case, indicating this is a global workspace variable. See Section 3.3.2.
stp	STP as defined in Table 3.1. If **stp** > 0 the algorithm assumes that the plant (**plt**), the processor noise (**wxx**) and the cost weighting matrices (**q1a**, **q2a**, and **ma**) are all defined for a *continuous* plant in feedback with the discrete compensator. If **stp** = 0 the algorithm assumes that the plant (**plt**), the processor noise (**wxx**) and the cost weighting matrices (**q1a**, **q2a**, and **ma**) are all defined for a *discrete* plant in feedback with the discrete compensator, and that the sampling period of the discrete plant is one STP of the multirate compensator

3.3.2 mropt_init

Format: **mropt_init**

Description: This script operates on the workspace variables defined in Table 3.2. It discretizes (if necessary) the plant, process noise and cost function; and defines a set of global workspace variables used by the optimization script (**mropt_optim**). This script must be rerun after making any changes in the workspace variables defined in Table 3.2, with exception to changes in the values of **cmp** or **cmpfree**.

Inputs: **mropt_init** is a script which uses the workspace variables defined in Table 3.2.

Outputs: mropt_init save it's output to the global workspace variables defined in Appendix A.

Comment: The global workspace variables defined by **mropt_init** are used by both the optimization script **mropt_optim** and by the M-File **calc_LQGcost**. All global variables generated by **mropt_init** begin with a capital letter. This helps to differentiate them from other workspace variables. A brief description of these global variables is given in Appendix A.

3.3.3 mropt_optim

Format: **mropt_optim**

Description: **mropt_optim** computes optimum values of the digital processor gains such that the cost function defined in Eqn. (3.18) is minimized.

Inputs: **mropt_optim** uses the global workspace variables generated by **mropt_init** in addition to the three workspace variables defined in Table 3.3.

Outputs: The script returns a value of the cost function, the gradient of the cost function with respect to the free compensator parameters, and the optimum values of the digital processor gains. These values are automatically displayed on the computer screen and simultaneously stored in the workspace variables defined in Table 3.4.

Comments:

1) As already noted, if the user modifies the problem definition by changing the values of the variables defined in Table 3.2, the script **mropt_init** must be rerun before running **nropt_optim**. The exceptions are changes to the variables defined in Table 3.3.
2) **mropt_optim** can be computationally intensive. The user can abort the optimization with a Control-C key sequence. Also see **mropt_extract**, Section 3.3.4

Table 3.3 Input Workspace Variables for **mropt_optim**

Variable	Description
cmp	This variable contains the compensator gains and is the *same* variable as described in Table 3.2. These are the starting values for the optimization and must stabilize all plant perturbations.
cmpfree	This matrix specifies which of the gains in **cmp** are fixed and is the *same* variable as described in Table 3.2.
dscale	During optimization the compensator gains are scaled using a variable named **dscale** to help improve the numerical accuracy of the search. They are scaled as follows $Xscaled = \mathbf{inv(dscale)} \cdot Xunscaled$ where *Xunscaled* is an unscaled vector containing the free compensator gains and *Xscaled* are those values scaled. Typically **dscale** is a diagonal matrix chosen so that all the elements in *Xscaled* have the same magnitude. The user can either manually set **dscale** or have **mropt_optim** set it automatically. If **dscale** exists *and* has the correct dimensions then **mropt_optim** uses that value of **dscale**, otherwise a new value for **dscale**, based on the current values in **cmp**, is automatically calculated. Typing "`clear dscale`" before running **mropt_optim** forces the routine to scale the problem based on current values in **cmp**.

Table 3.4 Output Workspace Variables for **mropt_optim**

Variable	Description
cmpf	This contains the optimized values of **cmp** after **mropt_optim** has completed the optimization.
jcost	This contains the value of the cost function after **mropt_optim** has completed the optimization.
djdp	This contains a scaled value of the gradient of the cost function with respect to the compensator values after **mropt_optim** has completed the optimization.

3.3.4 mropt_extract

Format: **mropt_extract**

Description: After aborting the script **mropt_optim**, usually with a Control-C key sequence, the script **mropt_extract** can extract the value of the "optimized" compensator at the last iteration.

Inputs: None. Can only be rerun after aborting **mropt_optim**.

Outputs: The optimized values of the digital processor gains at the last iteration before **mropt_optim** was aborted are displayed on the computer and stored in the workspace variable **cmpf**.

Comments: Sometimes it is necessary to abort **mropt_optim** before is has completed the optimization. The user might wish to rescale the free parameters by changing **dscale**, or might decide to free up additional gains in **cmp** by changing **cmpfree**. The user can restart the optimization from the last iteration by setting **cmp=cmpf** and rerunning **mropt_optim**.

3.3.5 calc_LQGcost

Format: **[jcost]=calc_LQGcost(j)**

Description: Calculates the discrete LQG cost assuming a sampling period of one **stp**. This routine uses the global workspace variables computed by **mropt_init**. Use this routine to find the minimum value of the cost function attainable by **mropt_optim**

Inputs: **j** = 0 or is undefined, then the routine calculate the sum of the LQG costs for all **Np** plant conditions corresponding to the cost $J_{discrete}$ given in Eqn. (3.18).

0<j<Np then the routine calculates the LQG cost for only the **j**th plant condition

Outputs: **jcost** The value of the LQG cost function

Comments: **calc_LQGcost** uses the Controls_Toolbox routines **dlqr** and **dlqe**. They require that **ra** in Table 3.2 be positive-definite.

NOTE: Additional M-Files used in support of those described in this section are described in Appendix B.

SECTION FOUR

SYNTHESIS EXAMPLE

4.0 PROBLEM DESCRIPTION

In this section we present an example design problem which utilizes the M-Files discussed in the previous section. Our objective is to design a multirate compensator for a lightly damped mass-spring-mass system.

The mass-spring-mass (MKM) system is shown in Figure 4.1. It consists of two masses connected by a spring and damper. The control inputs are the force inputs u_1 and u_2; the sensor outputs are the displacements x_1 and x_2; and the disturbance input is w. The sensor outputs are corrupted by discrete sensor noise (not shown on figure) with covariance v. Nominal values for the plant parameters, along with those for one known plant perturbation, are given in Table 4.1. These parameter values result in a system with two poles at the origin associated with x_1, and two lightly damped poles associated with x_2. The open-loop poles are listed in Table 4.1. A M-File which defines the system matrices for the MKM system is given in Appendix C.

The goal of our design is to increase the damping coefficient on the lightly damped poles to ζ=0.707 without shifting their natural frequency, and to move the two poles at the origin so that the x_1 response has a frequency of 0.825 rad/sec with a damping coefficient ζ=0.707.

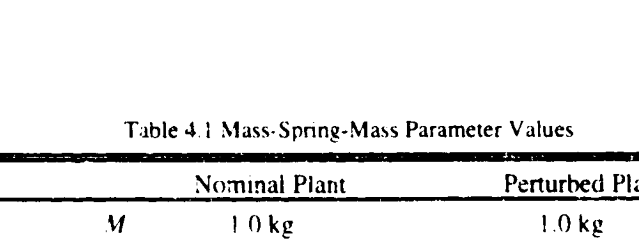


Figure 4.1 Mass-Spring-Mass System

Table 4.1 Mass-Spring-Mass Parameter Values

	Nominal Plant	Perturbed Plant
M	1.0 kg	1.0 kg
m	0.1 kg	0.2 kg
k	0.01 N/m	0.01 N/m
b	0.01 N·s/m	0.01 N·s/m
$E\{ w^T(t-\tau)w(t)\}$	$0.1\delta(\tau)N^2$	$0.1\delta(\tau)N^2$
$E\{ v^T(k+j)v(k)\}$	$0.001\delta(j)m^2$	$0.001\delta(j)m^2$
Open-Loop Poles	0, 0, -0.055± 3.316i	0, 0, -0.030±2.449i

4.1 COMPENSATOR DESIGN

Our compensator design followed the preceding steps. All computer inputs and outputs in the following paragraphs are in `this font`. **`Bolded text`** should be input by the user, `plain text` is returned by the computer, instructional comments are in *italic*.

Step 1: Select the weighting matrices for the quadratic cost function given in Eqn. (2.6) which characterize the desired closed-loop performance

The cost function given in Eqn. (2.6) is the sum of the costs associated with each plant perturbation. In our case we have two plants - the nominal and the perturbed plant. We determined the weighting matrices for our cost function by designing two continuous time LQ regulators which placed the closed-loop poles for the nominal and perturbed plants in the desired locations. The cost functions are

$$J_{\text{nominal}} = \mathrm{E}\{5.5x_1^2 + 2.2x_2^2 + 10000u_1^2 + 10u_2^2\} / 10.6 \tag{4.1a}$$

$$J_{\text{perturbed}} = \mathrm{E}\{6.5x_1^2 + 4.8x_2^2 + 10000u_1^2 + 10u_2^2\} / 8.0 \tag{4.1b}$$

where the costs have been scaled so that the LQR cost for each is unity. The cost function characterizing the performance of our multirate compensator is the sum of the these two cost functions. By minimizing (J_{nominal} + $J_{\text{perturbed}}$) the optimization routine will attempt to find a single compensator whose performance approaches that of the LQR designs for both the nominal and the perturbed plant. Of course we do not expect such a compensator to exist. Instead the resulting compensator will represent a compromise between good performance at the nominal plant condition and good performance at the perturbed condition.

Step 2: Select an appropriate compensator structure and sampling schedule based on the desired closed-loop performance

The closed-loop LQR system has a fast mode associated with x_1 and u_1 and a slow mode associated with x_2 and u_2. We selected a multirate compensator which capitalized on this structure. It consists of two *coupled* first-order loops, one from x_1 to u_1 and another x_2 to u_2. A block diagram of this structure is shown in Figure 4.2. The x_1 to u_1 loop is sampled and updated every 0.8 seconds while the x_2 to u_2 loop is sampled and updated four time as often, every 0.2 seconds. These sampling rates are approximately 10 time the desired closed-loop bandwith for each loop. In addition, the compensator accounts for a one-half sampling period computational delay. It's sampling schedule is shown in Figure 4.3.

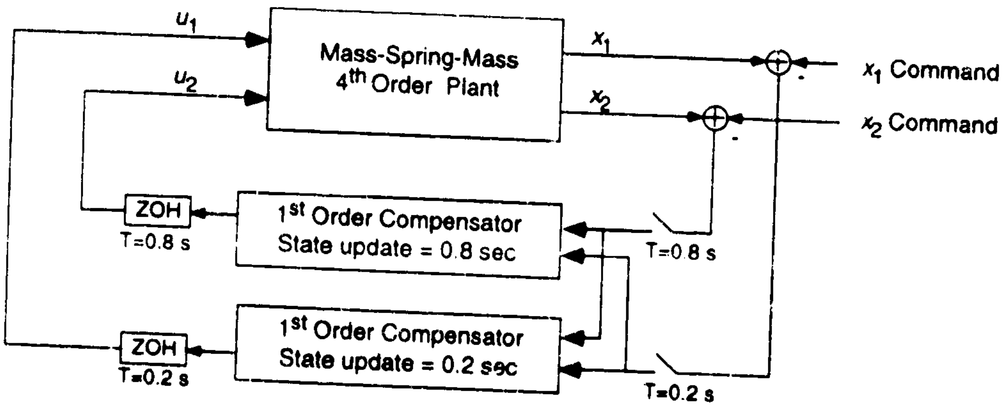


Figure 4.2 Block Diagram of Closed-Loop Multirate System

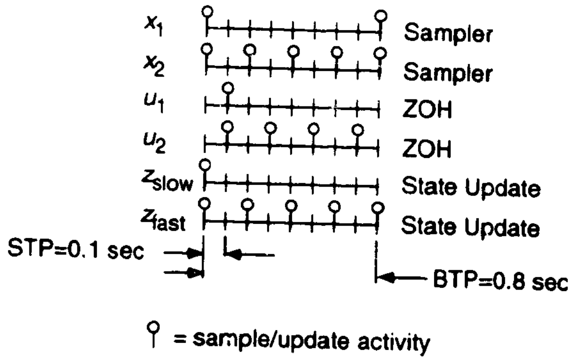


Figure 4.3 Sampling Schedule for Multirate Compensator

Step 3: Design a compensator with the sampling schedule and structure selected in Step 2 that stabilizes all plant perturbations. This will be the starting point for the optimization.

We designed a compensator using successive loop closures and root locus for the nominal plant. The compensator has the same structure and sampling schedule as the multirate design discussed in Step 2 except there is no input coupling and we did not account for computational delay. This compensator does however stabilize both the nominal and perturbed plants even with the computational delay and so can be used as the starting point for the optimization.

Refer to Reference 1 for an introduction to successive loop closures.

Step 4: Load the problem definition into Matlab's workspace variables defined in Table 3.2.

We defined the workspace variables using the Matlab script provided in Appendix D. To load the workspace variables for this example, type

```
»mropt_mkm
```

Step 5: Initialize the workspace variables and generate the necessary global variable using mropt_init

Type

```
»mropt_init    see Section 3.3.2
```

At this point you can compute the minimum value of the cost function as follows (see Section 3.3.5)

```
»calc_LQGcost  see Section 3.3.5
» ans =
     2.0108e+00    this is the minimum cost our design could achieve
```

Step 6: Calculate optimum values of the digital processor gains using mropt_optim

A partial output of the optimization follows

```
»mropt_optim
```

see Section 3.3.3

```
Automatically selecting dscale
```

a diagonal dscale is automatically selected if the variable ***dscale*** *does not exist, or is not compatible with the current compensator*

```
Calculate gradient only? (y or n)n
```

if we replied y the algorithm would compute only the gradient of the cost with respect to the free digital processor gains and the value of the cost for the current compensator gains in ***cmp***

f-COUNT = *the total number of function evaluations*
FUNCTION = *the value of the cost function at the current iteration. Refer to Matlab FMINU documentation*

```
f-COUNT   FUNCTION    STEP-SIZE       GRAD/SD  LINE-SEARCH
    2         176          0.01    -9.48e+07
-> Plant  1 unstable while calculating J
-> Plant  1 unstable while calculating J
-> Plant  1 unstable while calculating J
-> Plant  1 unstable while calculating J
-> Plant  1 unstable while calculating J
-> Plant  1 unstable while calculating J
-> Plant  1 unstable while calculating J
-> Plant  1 unstable while calculating J
```

The algorithm encountered a destabilizing solution. It automatically adjusts the step size and continues the optimization

```
  -9.9655e-01
  -1.6399e-02
   1.0030e+00
  -1.0676e-02
  -2.2977e-02
  -9.4363e-01
  -1.3703e-03
   1.0529e+00
   1.0030e+00
   1.0036e+00
   1.0529e+00
   1.0085e+00
        .
        .
        .
```

these are the values of the scaled digital processor gains at the current iteration

```
 313           4.62          14.2     2.45e-13   int_st
Optimization Terminated Successfully
Gradient less than options(2)
 NO OF ITERATIONS=314
  -8.0984e-01
  -1.6700e+01
   1.8697e+00
   2.2849e+01
   7.4219e-01
  -7.3479e+00
  -8.6882e+00
   7.9308e+00
   1.0706e+00
   3.8424e-03
   1.0367e+00
   1.4037e+00

****      Final Results      ****
       Gain          Gradient
   -----------   -----------
  -1.6197e-01  -4.6527e-06
  -1.6700e+01   6.6547e-04
   1.8697e+00  -2.8675e-04
   2.2849e+01   4.5050e-04  these are the unscaled digital processor gain values
   7.4219e-01   1.6576e-04  and their scaled gradient values
  -5.5109e+00  -8.9955e-04
  -8.6882e+00  -4.8353e-05
   7.9308e+00  -8.5171e-04
   8.5648e-02  -1.0838e-04
   1.9212e-03   6.9640e-04
   6.2202e-01   3.4133e-03
   1.4037e-01   3.2564e-04

Final cost = 4.62482            the optimum value of the cost function

Optimized compensator gains

cmpf =

   1.9212e-03             0    1.8697e+00  -8.6882e+00
            0    1.4037e-01    2.2849e+01   7.9308e+00
   8.5648e-02             0   -1.6197e-01   7.4219e-01
            0    6.2202e-01   -1.6700e+01  -5.5109e+00

Elapsed time: 1516.05 sec
```

4.2 DESIGN ANALYSIS

We looked at two criteria when evaluating our multirate design: 1) the final cost; and 2) the step response to a command input.

4.2.1 Final Cost

The final cost for the multirate compensator is calculated automatically by the optimization routine **mropt_optim**. The LQG cost is computed by the function **calc_LQGcost** in step 5. For our system

$$J_{\text{multirate}} = 4.6 \quad \text{and} \quad J_{\text{LQR}} = 2.0$$

4.2.2 Step Response

We obtained the response to a unit command step to x_1 as follows

```
»[a,b,c,d]=mkm; get the nominal plant
»b=[b zeros(4,1)];c=[c;[1 0 0 0]]; create a reference input
»d=[d zeros(4,1);0 0 -1];
»ur=[zeros(2,100);ones(1,100)];     generate a step input function for the input
                                    and compute the time domain response. see Section 3.2.1
»[y,t]=mrsim([a b;c d],4,cmpf,2,su,sy,sz,Stppbtp,1,[1 2],[5 3],ur,stp);
                          assuming cmpf, su, sy, sz, Stppbtp, and stp were previously defined and
                          that cmpf contains the optimized values of the digital processor gains
»plot(t,y([1 3],:))
```

We similarly obtained the step response for the LQR design. The results are shown in Figure 4.4.

As expected the performance of the multirate design does not match that of the LQ regulator. The LQ regulator was optimized for only the nominal plant. The multirate designs on the other hand represent a compromise, stabilizing both plants and providing "optimum" performance for both. In addition, the LQ design was a continuous full-state design whereas the multirate design was discrete second order and used only two state measurements.

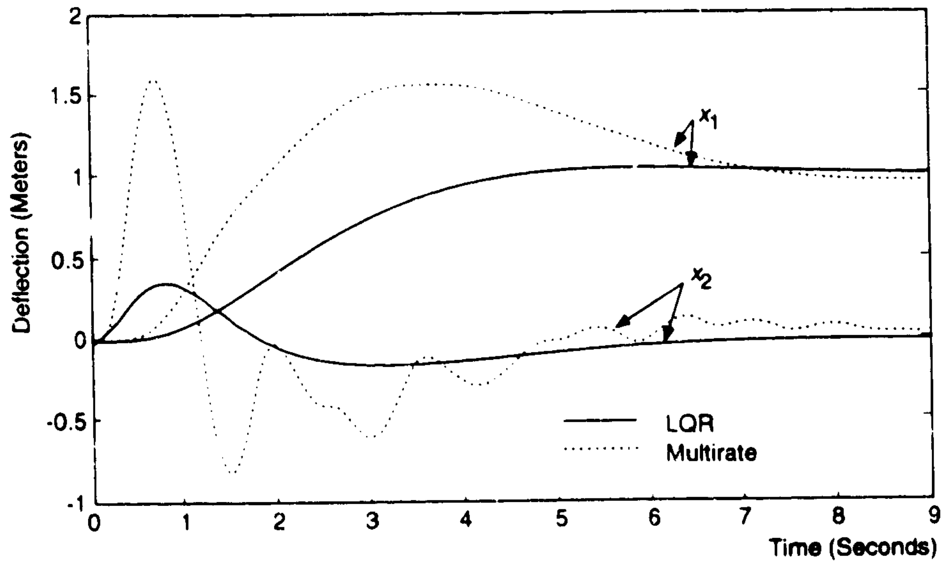


Figure 4.4 Response of MKM system to command step response to x_1

REFERENCES

1. Mason, G.S., "Multirate Compensator Synthesis and Analysis," Ph.D. Dissertation, UW, 1992

2. Mason, G.S. and Berg, M.C., "Reduced Order Multirate Compensator Synthesis," *AIAA Jour. Guid. Contr. and Dynamics*, Vol 15, No. 3 May-June 1992, pp. 700-706.

3. Mason, G.S. and Berg, M.C., "Robustness Analysis of a Multirate Flutter Suppression System," *1991 AIAA Guid. and Contr. Conf.*, accepted for publication in the *AIAA Jour. Guid., Contr. and Dynamics*.

4. Berg, M.C., Mason, G.S., and Yang, G.S., "A New Multirate Sampled-Data Control Law Structure and Synthesis Algorithm," *AIAA Jour. Guid. Contr. and Dynamics*, Vol 15, No. 5 September-October 1992, pp. 1183-1191.

APPENDIX A

GLOBAL WORKSPACE VARIABLES

The following global workspace variables are defined by **mropt_init**. Note that the first letter of every global variable is capitalized.

For eacʰ plant perturbation, the discrete closed-loop system can be written in the form:

$$x_i(m,n+1) = F_i x(m,n) + G_i u(m,n) + Ww_i \tag{A.1}$$

$$y_i(m,n) = H_i x(m,n) + Vw_i \tag{A.2}$$

$$u_i(m,n) = [S1(n)P(n)S2(n) + S3(\mathrm{n})]y_s(m,n) \tag{A.3}$$

where

$$F_i = \begin{bmatrix} \overline{F}_i & 0 \\ 0 & 0 \end{bmatrix}, G_i = \begin{bmatrix} \overline{G}_i & 0 \\ 0 & I \end{bmatrix}, W = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, H_i = \begin{bmatrix} \overline{H}_i & 0 \\ 0 & I \end{bmatrix}, V = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, P = \begin{bmatrix} D_{\underset{\sim}{z}} & C_{\underset{\sim}{z}} \\ B_{\underset{\sim}{z}} & A_{\underset{\sim}{z}} \end{bmatrix} \tag{A.4}$$

and w is a discrete zero mean white noise input with

$$E\{w_i(m,n)w_i^{\mathrm{T}}(m+j,n+k)\} = \{\delta(k)+\delta(j)\}R_i \tag{A.5}$$

The subscript i indicates the i^{th} plant condition and the overbarred matrices are the discretized plant matrices. The matrices $S1$, $S2$, and $S3$ are periodically time-varying switching matrices which model the multirate sampling scheme, and P is a matrix of the compensator's digital processor gains

The discrete cost function for the system in Eqns. (A.1-A.3) can be written as

$$J = \lim_{m\to\infty} \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathbf{Np}} \sum_{n=0}^{N-1} E\left\{ \begin{bmatrix} x_i(m,n) \\ u_i(m,n) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Q_{1i} & M_i \\ M_i^{\mathrm{T}} & Q_{2i} \end{bmatrix} \begin{bmatrix} x_i(m,n) \\ u_i(m,n) \end{bmatrix} \right\} \tag{A.6}$$

See Reference 1 for more information.

The global Matlab workspace variables are then defined as follows.

Plant Description:

Np = number of plant conditions

MF = $[F_1, F_2, \ldots, F_{\mathbf{Np}}]$	**NcF** = number of columns in F_i
MG= $[G_1, G_2, \ldots, G_{\mathbf{Np}}]$	**NcG**= number of columns in G_i
MW= $[W_1, W_2, \ldots, W_{\mathbf{Np}}]$	**NcW** = number of columns in W_i
MH= $[H_1, H_2, \ldots, H_{\mathbf{Np}}]$	**NcH**= number of columns in H_i
MV= $[V_1, V_2, \ldots, V_{\mathbf{Np}}]$	**NcV** = number of columns in V_i
MR= $[R_1, R_2, \ldots, R_{\mathbf{Np}}]$	**NcR** = number of columns in R_i

Compensator Description:

Nzhcmp = number of states in the periodically time-varying model of the multirate compensator

Stppbtp = the global version of **stppbtp**

S1k = $[S1(0), S1(0), ..., S1(N\text{-}1)]$ **Ns1** = the number of columns in $S1$

S2k = $[S2(0), S2(0), ..., S2(N\text{-}1)]$ **Ns2** = the number of columns in $S2$

S3k = $[S3(0), S3(0), ..., S3(N\text{-}1)]$ **Ns3** = the number of columns in $S3$

Ptv: abs(**Ptv**) = number of columns in the compensator gain matrix $[A_z, B_z; C_z, D_z]$. **Ptv** is positive if the digital processor gains are time-invariant and negative if they are periodically time-varying.

Cost Function Description:

MQ1= $[Q_{11}, Q_{12}, \dots Q_{1\,\mathbf{Np}}]$ **NcQ1=** number of columns in Q_{1i}

MQ2= $[Q_{21}, Q_{22}, \dots Q_{2\,\mathbf{Np}}]$ **NcQ2=** number of columns in Q_{2i}

MM= $[M_1, M_2, \dots M_{\mathbf{Np}}]$ **NcM=** number of columns in M_i

Miscellaneous Definitions:

Ppn , **Ppm**, and **Ppinit** : used internally

Ppfree = index to the "free" digital processor gains

Fcl = closed-loop state transitions matrix of the last computed ETIS system

Xc_global = value of the compensator gains at the last iteration

Unstable: If **Unstable** =0 the compensator gains at the current iteration stabilize the plants
Otherwise they destabilize the plant

APPENDIX B

SUPPORT M-FILES

B.1 MODELING SUPPORT M-FILES

B.1.1 mrsplit

Format: **[a, b, c, d] = mrsplit(P, n)**

Description: Separates the composite gain matrix **P** into its constituents, where

$$\mathbf{P} = \begin{bmatrix} \mathbf{d} & \mathbf{c} \\ \mathbf{b} & \mathbf{a} \end{bmatrix} \tag{B.1}$$

Inputs: **P** see Eqn. (B.1)
n the number of columns in **a**

Outputs: **a, b, c, d** the gain matrices in Eqn. (B.1)

B.1.2 mrmakesk

Format: **[s1k, ns1k, s2k, ns2k, s3k, ns3k]= mrmakesk(su, sy, sz, stppbtp)**

Description: The periodically time-varying representation of a multirate compensator in Eqn. (2.2) can be written as

$$\begin{bmatrix} y(m,n) \\ x(m,n+1) \end{bmatrix} = \begin{bmatrix} D(n) & C(n) \\ B(n) & A(n) \end{bmatrix} \begin{bmatrix} u(m,n) \\ x(m,n) \end{bmatrix} = \left\{ S1(n) \begin{bmatrix} D_z(n) & C_z(n) \\ B_z(n) & A_z(n) \end{bmatrix} S2(n) + S3(n) \right\} \begin{bmatrix} u(m,n) \\ x(m,n) \end{bmatrix} \tag{B.2}$$

where A_z, B_z, C_z and D_z are the digital processor gains and $S1$, $S2$ and $S3$ are switching matrices which model the compensator's sampling schedule. **mrmakesk** creates the periodically time-varying switching matrices given the sampling schedule description.

Inputs: **su, sy, sz, stppbtp** see Table 3.1

Outputs: **s1k** = [$S1(0)$ $S1(1)$ ... $S1($**stppbtp**-1$)$]
s2k = [$S2(0)$ $S2(1)$ ... $S2($**stppbtp**-1$)$]
s3k = [$S3(0)$ $S3(1)$ ... $S3($**stppbtp**-1$)$]
ns1k, ns2k and **ns3k** the number of columns in $S1(\cdot)$, $S2(\cdot)$ and $S3(\cdot)$ respectively

B.1.3 mrgetsk

Format: **[s1, s2, s3]= mrgetsk(k, s1k, ns1k, s2k, ns2k, s3k, ns3k)**

Description: Extracts the individual switching matrices for a given STP from **s1k**, **s2k** and **s3k**.

Inputs: **k** an integer specifying for which STP the switching matrices are to be extracted
s1k, ns1k, s2k, ns2k, s3k, ns3k see Section B 1.2.

Outputs: **s1, s2, s3** switching matrices corresponding to $S1($**k**$)$, $S2($**k**$)$ and $S3($**k**$)$ respectively. See Section B.1.2.

B.2 SYNTHESIS SUPPORT M-FILES

B.2.1 calc_djdp

Format: **djdp=calc_djdp(pguess. dscale)**

Description: Used by the routine **mropt_optim** to calculate the derivative of the cost J with respect to the free compensator parameters. See Section 2.3.

Inputs: **pguess** a vector of the values of the free digital processor gains at the current iteration
dscale see Table 3.3.

Comments: This routine uses the global workspace variables generated by **mropt_init**

B.2.2 calc_j

Format: **djdp=calc_j(pguess, dscale)**

Description: Used by the routine **mropt_optim** to calculate the cost J at the current iteration

Inputs: **pguess** a vector of the values of the free digital processor gains at the current iteration
dscale see Table 3.3.

Comments: This routine uses the global workspace variables generated by **mropt_init**

B.2.3 calc_L

Format: **[lacc]= calc_L(pk, f, g, w, h, q1, q2, m)**

Description: Called by **calc_djdp** to calculate the steady-state values of a Lagrange multiplier. (The multiplier is used to adjoin the stability equality constraints to the cost function.)

Inputs: **pk** gains for the periodically time-varying representation of the multirate compensator

$$\mathbf{pk} = [\,P(0), P(1), \ldots, P(N-1)] \text{ and } P(i) = \begin{bmatrix} D(i) & C(i) \\ B(i) & A(i) \end{bmatrix} \tag{B.3}$$

$D(i)$, $C(i)$, $B(i)$, and $A(i)$ are the state space matrices for the periodically time varying representation of the compensator. See Section 2.2 and Section B.1.2

f, g, w, h plant description matrices for the current plant perturbation corresponding to F_i, G_i, W and H_i in Eqn. (A.4).

q1, q2, m cost weighting matrices for the current plant perturbation corresponding to Q_{1i}, Q_{2i} and M_i in Eqn. (A.6).

Output: **lacc** periodically time varying lagrange multipliers where

$$\mathbf{lacc} = [\,\Lambda(1)\ \Lambda(2)\ \ldots\ \Lambda(N\text{-}1)\ \ \Lambda(0)]$$

and $\Lambda(i)$ is the lagrange multiplier for the i^{th} STP

Comments: This routine uses the global workspace variables generated by **mropt_init**

B.2.4 calc_P

Format: **[pk]=calc_P(p)**

Description: Called by **calc_j** and **calc_djdp** to compute the periodically time-varying representation of the multirate compensator

Inputs: **p** the digital processor gains in the form

$$\mathbf{p} = [p(0), p(1),...,p(N\text{-}1)] \text{ where } p(i) = \begin{bmatrix} D_z(i) & C_z(i) \\ B_z(i) & A_z(i) \end{bmatrix}$$

Outputs: **pk** matrix of the compensator gains for the periodically time-varying representation of the multirate compensator

$$\mathbf{pk} = [P(1), P(2), P(3),... P(N)] \text{ where } P(i) = S1(i)p(i)S2(i) + S3(i)$$

See Eqn. (A.2) or Section B.1.2.

Comments: This routine uses the global workspace variables generated by **mropt_init**

B.2.5 calc_X

Format: **[xacc]= calc_X(pk, f, g, w, h, v, r)**

Description: Called by **calc_djdp** and **calc_j** to calculate the steady-state covariance values of the closed-loop system

Inputs: **pk** matrix of the compensator gains for the periodically time-varying representation of the multirate compensator, see Section B.2.4.

f, g, w, h plant description matrices for the current plant perturbation corresponding to F_i, G_i, W, H_i and V in Eqn. (A.4) respectively

r process and sampling noise covariances for the current plant perturbation corresponding to R_i in Eqn. (A.5).

Comments: This routine uses the global workspace variables generated by **mropt_init**

B.2.6 disc_cost

Format: **[q1d, q2d, md]=disc_cost(plt, nplt, stp, q1, q2, m)**

Description: Computes the discrete cost function weighting matrices such that the cost associated with a continuous plant in feedback with a single-rate compensator is identical to the cost associated with the discretized plant in feedback with the single-rate compensator. The continuous cost function is given in Eqn. (2.6); the discrete cost function is given by

$$J_{discrete} = \lim_{m \to \infty} \frac{1}{N} \sum_{n=0}^{N-1} E\left\{ \begin{bmatrix} x(m,n) \\ u(m,n) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Q_{1d} & M_d \\ M_d^{\mathrm{T}} & Q_{2d} \end{bmatrix} \begin{bmatrix} x(m,n) \\ u(m,n) \end{bmatrix} \right\} \qquad (B.4)$$

disc_cost assumes the control input is updated using a zero-order-hold.

Inputs: **plt, nplt, stp** see Table 3.1

q1, q2, m the continuous cost function weighting matrices in Eqn. (2.6)

Outputs: **q1d, q2d, md** the discrete weighting matrices in Eqn. (B.4)

B.2.7 disc_noise

Format: **[w]=disc_noise(wxx, plt, nplt, stp)**

Description: Computes the covariance of a discrete-time white noise process disturbance such that the expected value of the states of a continuous plant, and the expected value of the states of the discretized plant are equivalent at the sampling instances. Thus, given the continuous plant

$$\dot{x}(t) = Ax(t) + G_w w(t), \text{ where } E(w(t)w^T(t+\tau)) = \delta(\tau)\mathbf{wxx}$$

and the plant discretized with a zero-order-hold at a sampling period T

$$x_d(m,n+1) = A_d x_d(m,n) + w_d(m,n), \text{ where } E(w_d(m,n)w_d^{\mathrm{T}}(m+k,n+j)) = (\delta(k)+\delta(j))\mathbf{w}$$

then $E\{x_d(m,n)x_d^{\mathrm{T}}(m,n)\} = E\{x((mN+n)T)x^{\mathrm{T}}((mN+n)T)\}$

Inputs: **wxx** PSD of the continuous processor noise

plt, nplt, stp see Table 3.1. The inputs to **plt** are assumed to be only the process noise

Outputs: **w** PSD of the discrete process noise

B.2.8 dlyap2

Format: **x=dlyap2(a, c, method)**

Description: Solves the discrete lyapunov equation

$$\mathbf{x} = \mathbf{a}\cdot\mathbf{x}\cdot\mathbf{a}^{\mathrm{T}} + \mathbf{c} \tag{B.5}$$

Inputs: **a, c** system matrices in Eqn. (B.5)

if **method** = 'Bartels' then the algorithm solves Eqn. (B.5) using Bartels method

otherwise Eqn. (B.5) is solved using partial sums

Outputs: **x** the solution to the lyapunov equation

B.2.9 get_cost

Format: **[q1, q2, m]=get_cost(i)**

Description: Extracts the cost function weighting matrices $Q1$, $Q2$ and M for the **i**th plant perturbation from the global workspace variables **MQ1**, **MQ2**, and **MM**. See Appendix A.

Inputs: **i** an index to the desired plant perturbation

Outputs: **q1, q2, m** the cost weighting matrices corresponding to $Q1$, $Q2$ and M respectively

Comments: This routine uses the global workspace variables generated by **mropt_init**

B.2.10 get_plt

Format: **[f, g, w, h, v, r]=get_plt(i)**

Description: Extracts the plant matrices F_i, G_i, W, H_i, V, and R_i for the **i**th plant perturbation from the global workspace variables **MF**, **MG**, **MW**, **MH**, **MV** and **MR**. See Appendix A.

Inputs: **i** an index to the desired plant perturbation

Outputs: **f**, **g**, **w**, **h**, **v**, **r** plant description matrices for the **i**th plant perturbation corresponding to F_i, G_i, W, H_i, V, and R_i respectively

Comments: This routine uses the global workspace variables generated by **mropt_init**

B.2.11 get_ppfree

Format: **ppfree = get_ppfree(cmpfree, ncmp)**

Description: Called by **mropt_optim** to determine which compensator gains will be optimized

Inputs: **cmpfree**, **ncmp** see Table 3.1.

Outputs: **ppfree** vector whose elements point to the free elements in **cmp**

B.2.12 get_sk

Format: **[s1, s2, s3]=get_sk(k)**

Description: Extracts the switching matrices $S1(k)$, $S2(\mathrm{k})$ and $S3(k)$ for the **k**th STP from the global workspace variables **S1k**, **S2k**, and **S3k**. See Appendix A.

Inputs: **k** indicates the **k**th STP of the current BTP. NOTE: **k** = 1, 2, ... , **stppbtp**

Outputs: **s1**, **s2**, **s3** switching matrices for the **k**th STP corresponding to $S1(\mathbf{k})$, $S2(\mathbf{k})$ and $S3(\mathbf{k})$ respectively

Comments: This routine uses the global workspace variables generated by **mropt_init**

B.2.13 make_noise

Format: **[r]=make_noise(rw, rv)**

Description: Creates a compound noise covariance matrix from **rw** and **rv**

Inputs: **rw**, **rv** the process and sensor noise covariance matrices respectively

Outputs: **r** compound noise matrix of the form

$$\mathbf{r} = \begin{bmatrix} \mathbf{rw} & \mathbf{0} \\ \mathbf{0} & \mathbf{rv} \end{bmatrix}$$

B.2.14 make_plt

Format: **[f, g, w, h, v, nzcmp]=make_plt(pltc, nplt, ncmp, stp)**

Description: Creates the compound plant matrices F_i, G_i, W and V for the current plant perturbation. See Appendix A.

Inputs: **pltc** has the same form as **plt** in Table 3.1 except it describes only a single plant perturbation
nplt, ncmp, stp See Table 3.1.

Outputs: **f, g, w, h, v** compound plant matrices F_i, G_i, W and V for the current plant perturbation respectively
nzcmp the number states in the periodically time-varying representation of the multirate compensator

B.2.15 mropt_check

Format: **mropt_check**

Description: **mropt_check** performs elementary error checking on the data in Table 3.2 and set the variable **err**=1 if it finds an error in the data. **mropt_check** is called by **mropt_init.**

Inputs: None

Outputs: None

B.2.16 mropt_global

Format: **mropt_global**

Description: Called by **mropt_init** to define the global workspace variables

Inputs: None

Outputs: None

B.2.17 mropt_fminu

Format: **[x, OPTIONS] = mropt_fminu(FUN, x, OPTIONS, GRADFUN, P1, P2, P3, P4, P5, ... P6, P7, P8, P9, P10)**

Description: A modified version of **FMINU** from the Optimization_Toolbox. This version automatically reduces the search step size when a destabilizing solution is encountered.

Inputs: See **FMINU**

Outputs: See **FMINU**

Comments: Requires the Optimization_Toolbox

APPENDIX C

M-FILE MKM

The following M-File creates the state space matrices for the Mass-Spring-Mass system of Section 4.0

```
function [a,b,c,d]=mkm(m1,m2,k,b)
%[a,b,c,d]=mkm(m1,m2,k,b)

if nargin~=4  %nominal plant
  m1=1;
  m2=.1;
  k=1;
  b=0.01;
end;

t=(1/m1)+(1/m2);

a=[0   1   0     0
   0   0   k/m1 b/m1
   0   0   0     1
   0   0  -k*t  -b*t];

b=   [0       0
      1/m1   0
      0       0
     -1/m1   1/m2];

c=eye(4);
d=zeros(4,2);
```

APPENDIX D

SCRIPT MROPT_MKM

The following script defines the workspace variables in Table 3.2 for the example problem of Section 4.0

```
% Script: mropt_mkm.m
% Example script for creating input data for mropt
%
% Creates the following data required by mropt_init
%   plt,nplt,wxx,rv,q1a,q2a,ma
%   ucol,ncol,srow,crow
%   sz,su,sy,cmp,stp,Stppbtp,cmpfree

% G. Mason, U.W. 1992

% ----- Continuous plant description -----

Np=2;          % The number of plant conditions
nplt=4;        % The number of states in the plants
plt=[];        % Clear the variable which holds the plant data

% Plant #1
[a,b,c,d]=mkm(1,0.1,1,0.01);
plt = [a b; c d];
% Plant #2
[a,b,c,d]=mkm(1,0.2,1,0.01);
plt = [plt [a b;c d]];

% Pointer to rows and columns in the plant
ucol=[1 2];       % control input
ncol=2;           % process noise input
srow=[1 3];       % sensor output from the plant
crow=[1 2 3 4];   % criterion output from the plant

% Continuous process noise PSD
% wxx is a compound matrix like plt, it contains wxx for plant #1 and #2
wxx=[.1  .1];

% Discrete sensor noise PSD
% with same compound form as plt
rv=eye(2)*0.001;
rv=[rv rv];

% ----- Continuous Cost weighting matrices -----
% Output criterion weights
% Use 1e-8 instead of 0 so that qa1 & qa2 are positive definite
% The synthesis algorithm would accept semi-definite q1a & q2a
% but calc_LQGcost will not

q1a1=diag([5.5 1e-8 1e-8 2.2]);
q1a2=diag([6.5 1e-8 1e-8 4.8]);
q1a=[qa1/10.6 qa2/8.0];

% Control input weights
q2a=diag([1e4 1e1]);
```

```
q2a=[ra/10.6 ra/8.0];

% Cross weighting (ma)
% if there is no cross weighting it can be left undefined
clear ma

% ----- Compensator description -----
% Digital processor gains from successive loop closures design
cmp=[ 0.5  0      1     0
       0    0 1    0     1
       0.08 0     -0.2   0
       0    0.6    0   -0.75];

% Sampling schedule
su=[0 0;1 1;0 0;0 1;0 0;0 1;0 0;0 1];
                  % Compensator output updating w/ delay
sy=[8 2];         % Sensor input sampling, multiplexed
sz=[8 2];               % Compensator state update
stp = .1;               % The shortest sampling period
Stppbtp = 8;            % The number of stp's in one BTP

% Free compensator gains
cmpfree=[1 0 1 1
         0 1 1 1
         1 0 1 1
         0 1 1 1];

% clear temporary variables
clear a b c d
```