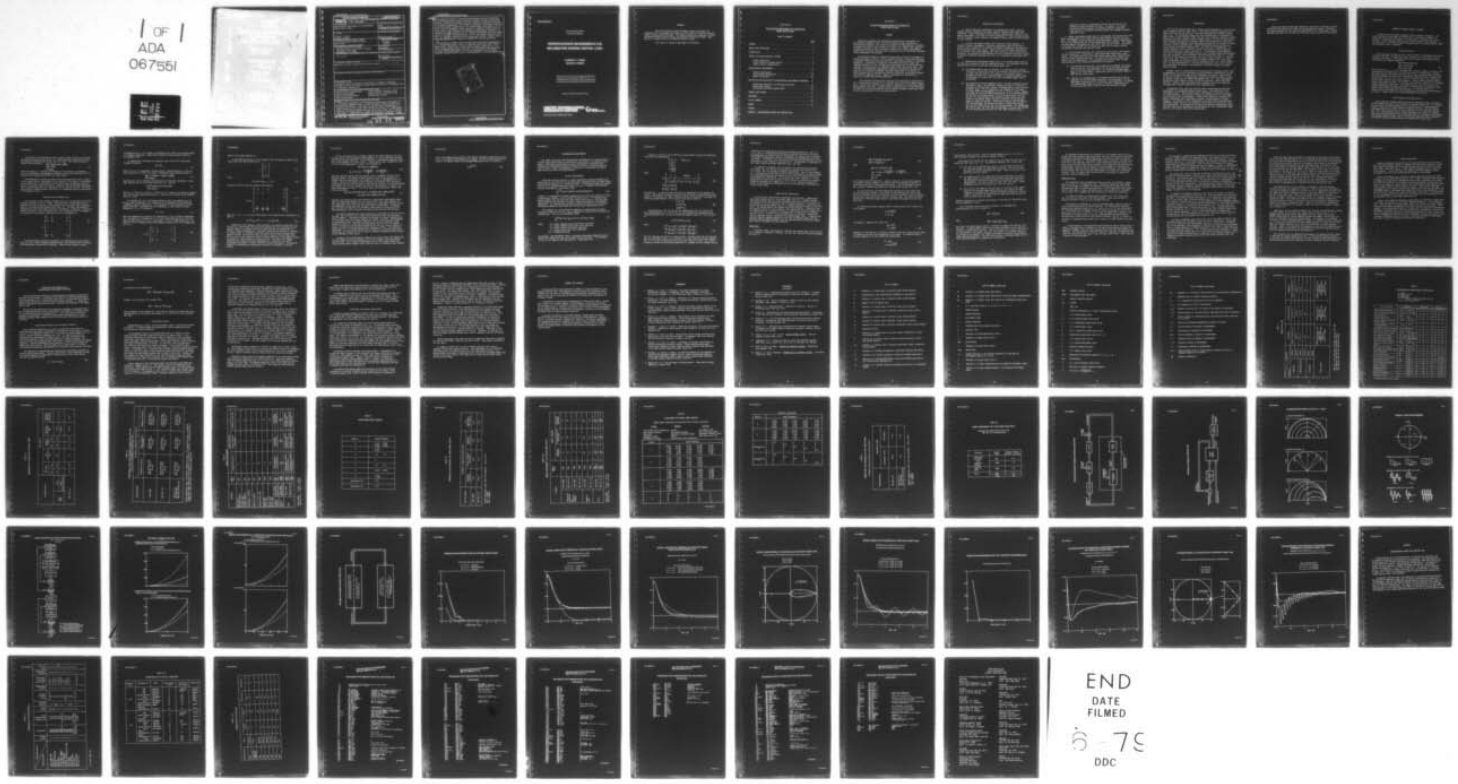
AD-A067 551 UNITED TECHNOLOGIES RESEARCH CENTER EAST HARTFORD CONN F/G 9/2
MICROPROCESSOR REQUIREMENTS FOR IMPLEMENTING MODERN CONTROL LOG--ETC(U)
MAR 79 F A FARRAR, R S EIDENS F49620-78-C-0017
UNCLASSIFIED UTRC/R79-944258-2 AFOSR-TR-79-0449 NL

1 OF 1
ADA
067551



END
DATE
FILMED
6-79
DDC

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFOSR-TR- 79-0449	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Microprocessor Requirements for Implementing Modern Control Logic	5. TYPE OF REPORT & PERIOD COVERED: Final, 1 February 1978 - 31 January 1979
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Florence A. Farrar, Richard S. Eidens	8. CONTRACT OR GRANT NUMBER(s): F49620-78-C-0017
9. PERFORMING ORGANIZATION NAME AND ADDRESS: United Technologies Research Center, Silver Lane, East Hartford, Connecticut 06108	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ~~78-00445~~ 2304/A1
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research, Bolling Air Force Base, Washington D. C. 20332	12. REPORT DATE: March 1979
	13. NUMBER OF PAGES: 71
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): 61102F	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Preliminary results presented at the Society of Automotive Engineers Aerospace Control and Guidance Systems Committee Meeting on 28 September 1978 in Cambridge, Massachusetts.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Control Implementation
Accuracy Requirements
Computational Requirements
Memory Requirements
Linear Quadratic Gaussian Control
Modern Control on Microprocessors
Microprocessor Control

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Analytical procedures for establishing microprocessor requirements for multivariable feedback control of linear stochastic dynamic systems were developed and evaluated. Key issues in microprocessor implementation of modern estimation and control logic include (1) accuracy, (2) computational capability including arithmetic as well as interface speed, and (3) memory requirements. Compatibility of current microprocessor technology with the established implementation requirements was assessed. —> next page (over)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

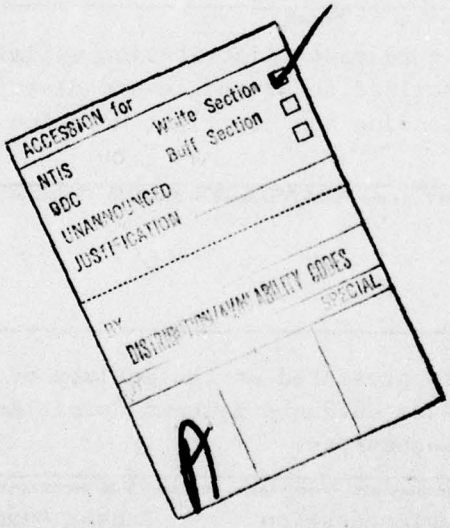
UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 04 12 062

cont.

20. The analytical method to establish microprocessor accuracy requirements for closed-loop digital control of linear stochastic systems is based upon defining a quadratic performance index that provides a quantitative measure of performance degradation. This method is employed to determine closed-loop system time response sensitivity to (1) microprocessor word length and (2) system structure (standard, Jordan canonical or companion forms) within the control logic. The technique for determining computational capability requirements includes procedures for (1) defining the maximum controller sample time as a function of system dynamics and (2) calculating controller arithmetic and interface computation time per sampling interval. Memory requirements are delineated as a function of system model and microprocessor.

The developed procedures were evaluated and illustrated by application to (1) a second-order system and (2) a linearized fifth-order F100 turbofan engine model. Results were verified using a digital simulation of a continuous system/microprocessor controller. Microprocessors considered were the Intel 8080 microprocessor (an 8 bit microprocessor) and the Digital Equipment Corporation (DEC) LSI 11/2 microprocessor (a 16 bit microprocessor).



R79-944258-2

FINAL TECHNICAL REPORT
REPORT R79–944258–2

MICROPROCESSOR REQUIREMENTS FOR IMPLEMENTING MODERN CONTROL LOGIC

FLORENCE A. FARRAR
RICHARD S. EIDENS

Research sponsored by the Air Force Office of Scientific Research (AFSC), United States Air Force, under contract No. F49620–78–C–0017. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

Approved for public release; distribution unlimited

UNITED TECHNOLOGIES RESEARCH CENTER

UNITED TECHNOLOGIES

EAST HARTFORD, CONNECTICUT 06108

79–03–198–8

FOREWORD

This final technical report documents research performed from 1 February 1978 to 31 January 1979 under Air Force Office of Scientific Research (AFOSR) Contract F49620-78-C-0017. The research program was conducted at United Technologies Research Center (UTRC), East Hartford, Connecticut 06108. Major Charles L. Nefzger served as the AFOSR Scientific Officer.

This report is issued as UTRC Report R79-944258-2.

Microprocessor Requirements for Implementing Modern Control Logic

TABLE OF CONTENTS

	Page
SUMMARY	1
RESULTS AND CONCLUSIONS	2
INTRODUCTION	4
CONTROL OF LINEAR STOCHASTIC SYSTEMS	6
System Description	6
Linear Quadratic Gaussian Control	6
Digital Control Implementation	7
MICROPROCESSOR REQUIREMENTS	12
Accuracy Requirements	12
Computational Requirements	14
Memory Requirements	20
APPLICATION AND EVALUATION OF MICROPROCESSOR REQUIREMENT PROCEDURES	21
Closed-Loop Simulation for Verifying Procedures	21
Second-Order System	22
Fifth-Order Gas Turbine Engine Model	24
CURRENT UTRC PROGRAM	26
REFERENCES	27
LIST OF SYMBOLS	29
TABLES	33
FIGURES	
APPENDIX - MICROPROCESSOR SURVEY AND COMPUTER CODE	

R79-944258-2

Microprocessor Requirements for Implementing Modern Control Logic

SUMMARY

Analytical procedures for establishing microprocessor requirements for multivariable feedback control of linear stochastic dynamic systems were developed and evaluated. Key issues in microprocessor implementation of modern estimation and control logic include (1) accuracy, (2) computational capability including arithmetic as well as interface speed, and (3) memory requirements. Compatibility of current microprocessor technology with the established implementation requirements was assessed.

The analytical method to establish microprocessor accuracy requirements for closed-loop digital control of linear stochastic systems is based upon defining a quadratic performance index that provides a quantitative measure of performance degradation. This method is employed to determine closed-loop system time response sensitivity to (1) microprocessor word length and (2) system structure (standard, Jordan canonical or companion forms) within the control logic. The technique for determining computational capability requirements include procedures for (1) defining the maximum controller sample time as a function of system dynamics and (2) calculating controller arithmetic and interface computation time per sampling interval. Memory requirements are delineated as a function of system model and microprocessor.

The developed procedures were evaluated and illustrated by application to (1) a second-order system and (2) a linearized fifth-order F100 turbofan engine model. Results were verified using a digital simulation of a continuous system/microprocessor controller. Microprocessors considered were the Intel 8080 microprocessor (an 8 bit microprocessor) and the Digital Equipment Corporation (DEC) LSI 11/2 microprocessor (a 16 bit microprocessor).

RESULTS AND CONCLUSIONS

1. Digital electronic requirements for implementing optimal stochastic feedback controls designed using linear quadratic Gaussian (LQG) theory were delineated. These requirements include accuracy, computational capability, and memory requirements. The effect on these requirements of various system representations (standard, Jordan canonical and companion forms) within the control logic was investigated.

2. Accuracy requirements depend upon closed-loop system dynamics. To establish accuracy requirements a quadratic performance index was defined to provide a direct indication of transient performance degradation associated with finite microprocessor word lengths. Results show that performance degradation as a function of word length depends upon model structure within the controller. Therefore, the performance index approach is used to determine the model structure to minimize performance degradation as well as to determine microprocessor word length.

3. Computational requirements depend upon (1) the maximum sample time and (2) controller computation time per sampling interval. The computation time per sampling interval must be less than the maximum sample time.

(a) The maximum sample time is a function of closed-loop system dynamics. Plotting the z-plane root locus of the discrete controller as a function of sample time was shown in this study to be an effective way to determine the effect on system performance of different sample times. By analyzing the z-plane root-locus plot the maximum sample time is established.

(b) The computation time per sampling interval includes arithmetic and I/O computation times. The arithmetic computation time is a function of the number of system states, inputs, and measurements as well as microprocessor speed. The I/O computation time depends upon the number of system inputs and measurements as well as interface speed. Computation times for the Intel 8080 microprocessor (an 8 bit microprocessor), the DEC LSI 11/2 microprocessor (a 16 bit microprocessor), and the DEC LSI 11/2 microprocessor with hardware multiply option were tabulated for the generic linear system as a function of system order, number of inputs and outputs. These tables indicate that the DEC LSI 11/2 microprocessor with hardware multiply is significantly faster than the Intel 8080 and DEC LSI 11/2 microprocessors. For implementing LQG control logic the Intel 8080 microprocessor is faster than the DEC LSI 11/2 microprocessor because the Intel 8080 software multiply

subroutine (8 bits) is significantly faster than the DEC LSI 11/2 software multiply subroutine (16 bits). In addition, the Jordan canonical structure requires the minimum arithmetic computation time; whereas, the standard structure requires the maximum arithmetic computation time.

4. Memory requirements depend upon the system model and microprocessor code. Memory requirements for the Jordan canonical and companion structures are the same. The standard structure requires the maximum memory. Memory requirements for the Intel 8080 and DEC LSI 11/2 microprocessors were tabulated. These tables show that memory requirements for LQG control logic are not severe. For example, a controller for a system with 10 states, 10 inputs and 10 outputs requires less than 1K of RAM and less than 1K of PROM.

5. The developed procedures were validated by application to two systems. The first system was a single-input, single-measurement second-order linear system. The second plant was an F100 gas turbine engine model linearized about sea-level static military operation. The engine model has five states, four inputs, and five measurements. The procedures were verified using a closed-loop simulation of the continuous plant/discrete controller.

(a) An Intel 8080 microprocessor can be employed to implement the second-order controller. The requirements for the second-order controller are (1) 8-bit word length accuracy, (2) minimum and maximum sample times of 2.73 and 700 msec, respectively, and (3) 347 and 21 words of PROM and RAM, respectively.

(b) A DEC LSI 11/2 microprocessor with hardware multiply can be used to implement the LQG F100 engine controller. The requirements for the F100 engine controller are (1) 12-bit word length accuracy, (2) minimum and maximum sample times of 9.68 and 25.0 msec, respectively, and (3) less than 200 words of RAM and less than 200 words of PROM.

INTRODUCTION

Over the past several years use of modern control methodology -- in particular, linear quadratic Gaussian (LQG) theory -- has gained increased recognition as an effective design tool for control of nonlinear multivariable stochastic systems (Refs. 1-8). The referenced studies have been conducted under a combination of AFOSR, Office of Naval Research (ONR), Air Force Aero Propulsion Laboratory (AFAPL), NASA-Lewis and Pratt & Whitney Aircraft (P&WA) support. In these as well as many other aerospace applications the primary impetus for application of modern LQG control concepts is improved system performance combined with the advent of digital electronic control implementation. Digital electronics provide the means by which complex controllers associated with LQG theory can be implemented. The current trend both within as well as outside the aerospace controls community toward increased use of digital electronics -- in particular, microprocessors -- will lead to increased use of modern control logic including system identification, modeling, estimation, and multivariable control methodologies (Ref. 9). In addition, use of microcomputer controllers will lead, in many instances, to reduced control cost (Refs. 10 and 11), lighter and smaller controls (Ref. 12), lower power requirements and integrated circuit reliability (Ref. 13). Recent studies (Ref. 14) have demonstrated that existing microprocessors can be used to implement algorithms for parameter identification of relatively simple, low-order dynamic systems.

However, prior to widespread use of microprocessors for modern control logic implementation, key issues associated with microprocessor implementation of LQG control and estimation concepts must be addressed and resolved. These issues include (1) accuracy, (2) computational capability including arithmetic as well as interface speed, and (3) memory requirements (Ref. 15). These requirements depend upon system dynamics as well as upon the particular control algorithm employed. Defining these requirements will establish criteria for selecting the appropriate computer system for control implementation.

Consequently, this study was directed toward establishing microprocessor requirements for implementing modern control logic. The objective of this investigation was to define microprocessor requirements of LQG control logic for linear systems. An analytical procedure to determine accuracy requirements for LQG control logic was developed. Computational capability, interface and memory requirements were established for generic representations of system dynamics. The effects on these requirements of various system representations (standard, Jordan canonical, and companion forms) within the LQG control logic were investigated. To evaluate and illustrate the developed methodology, microprocessor LQG control requirements were delineated for (1) a second-order model and (2) a fifth-order F100 turbofan engine model.

Because the linear system model employed in this study is general in nature, results of this study are not unique to any one particular system. Therefore, the results obtained in this study have applicability to a broad range of control problems with which the Air Force and the other military services are concerned.

CONTROL OF LINEAR STOCHASTIC SYSTEMS

In this section the linear stochastic control problem is presented. The linear stochastic system model is defined first. This model is general in nature and is therefore applicable to a broad range of estimation and control problems. Linear quadratic Gaussian (LQG) control design is then outlined. In the final part of this section digital implementation of the multivariable control and estimation logic is discussed.

System Description

The system model is shown in Fig. 1 with provision for estimation and control algorithms included. The open-loop system consists of a linear plant, control actuators, and sensors. Open-loop linear system dynamics are described by the differential and algebraic equations

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + B\,u(t) + \xi(t) \\ y(t) &= C\,x(t) + D\,u(t) \\ z(t) &= E\,x(t) + \eta(t) \end{aligned} \tag{1}$$

where the vectors x, u, y, and z represent the n states, m inputs, p outputs, and ℓ measurements, respectively. Note that the vector x includes plant, actuator, and sensor states. The random process vectors ξ and η represent white zero-mean Gaussian n-dimensional process and ℓ-dimensional measurement noise, respectively. The constant A (n x n), B (n x m), C (p x n), D (p x m) and E (ℓ x n) matrices define linear time-invariant system dynamics. The dot notation denotes differentiation with respect to time. The initial state vector is assumed to be a Gaussian random variable with known mean. The random vectors x(0), ξ(t), and η(t) are assumed independent with known covariances. The statistical properties of these random vectors define the system inaccuracies. The process noise (ξ(t)) models actuator uncertainties, plant disturbances, and system-to-system parameter variations. Sensor noise (n(t)) models measurement inaccuracies.

Linear Quadratic Gaussian Control

Linear quadratic Gaussian theory provides the analytical tools to design an optimal feedback control for the stochastic system described by Eq. (1). Solution of the overall LQG problem separates into the solution of (1) linear quadratic regulator (LQR) and (2) linear Gaussian estimation problems. The key theorem that demonstrates this property is often referred to as the separation theorem (Ref. 16). The stochastic control design procedures then involve (1) deterministic multivariable feedback control using LQR theory, (2) stochastic filter design using Kalman estimation logic, and (3) closed-loop regulation based on feedback of estimated system variables through the deterministic control logic.

The solution to the stochastic linear optimal control problem is well-known and the details can be found in Ref. 16. Under appropriate controllability and observability conditions the optimal input u* exists. The filter and control dynamics are described by

$$\dot{\hat{x}}(t) = F\hat{x}(t) + H(z(t) - E\hat{x}(t))$$
$$u^*(t) = G\hat{x}(t) \qquad (2)$$
$$F \triangleq A + BG$$

where the notation (^) denotes the estimate of the variable in parentheses, G (m x n) represents the deterministic feedback control gain matrix and H (n x ℓ) represents the steady-state Kalman filter gain matrix.

The deterministic control and Kalman filter gain matrices (G and H, respectively) depend upon (1) the dynamics of the system, (2) the levels of uncertainty in the system, and (3) performance criteria that specify satisfactory time evolution of the system inputs and outputs. However, since the problem assumptions lead to constant gain matrices, these gains can be computed off-line. The resulting stochastic feedback control structure to be implemented using microprocessors is illustrated in Fig. 2.

Digital Control Implementation

The structure of the stochastic feedback control to be implemented with microprocessor is shown in Fig. 2. Figure 2 indicates that a system model must be employed within the controller. Many mathematically equivalent model representations of the linear system dynamics exist (Ref. 17). The model representation employed will affect the digital electronic requirements. Generally, for physical systems the matrix F will be standard form. For the standard form all elements of the matrix F are assumed to be nonzero and F is represented by

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdot & \cdot & f_{1n} \\ f_{21} & f_{22} & \cdot & \cdot & f_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ f_{n1} & f_{n2} & \cdot & \cdot & f_{nn} \end{bmatrix}. \qquad (3)$$

To reduce digital electronic requirements for implementing the LQG control logic (Eq. (2)), state coordinates employed within the controller may be different from the physical system state coordinates. Several forms obtained by state

transformation (i.e., by a change of coordinates) will reduce the required number of arithmetic operations. These forms include the Jordan canonical form and the companion form.

To change state coordinates the estimated state vector $\hat{x}$ is transformed through the equation

$$\hat{w} = T^{-1}\hat{x} \tag{4}$$

where T is an n x n nonsingular constant matrix. Substituting Eq. (4) into the state equations for $\hat{x}$ (Eq. (2)) leads to transformed state equations given by

$$\begin{aligned} \dot{\hat{w}}(t) &= T^{-1}FT\,\hat{w}(t) + T^{-1}H\left[z(t) - ET\hat{w}(t)\right] \\ u(t) &= GT\hat{w}(t). \end{aligned} \tag{5}$$

Equations (2) and (5) represent the same system in different coordinates. Note, also, that Eq. (5) is of the same form as Eq. (2). That is,

$$\begin{aligned} \dot{\hat{w}}(t) &= F_T\,\hat{w}(t) + H_T\left(z(t) - E_T\hat{w}(t)\right) \\ u(t) &= G_T\hat{w}(t) \end{aligned} \tag{6}$$

where $F_T = T^{-1}(F)T$, $H_T = T^{-1}H$, $E_T = ET$ and $G_T = GT$. However, the matrix F_T depends upon the selected transformation matrix T. Note that T=I results in the standard form, i.e., F_T=F.

To transform Eq. (2) to the Jordan canonical form, the eigenvalues of the matrix F must be employed. For convenience it is assumed here that the matrix F possesses distinct eigenvalues. This assumption is quite reasonable for most physical systems. Associated with each eigenvalue λ_i there exists a nonzero eigenvector ω_i (n x 1) defined by

$$F\omega_i = \lambda_i\omega_i. \tag{7}$$

When the eigenvalues are distinct the eigenvectors ω_i form a linearly independent set. The matrix T to transform Eq. (2) to the Jordan canonical form is the non-singular modal matrix of F (i.e., the columns of T are the eigenvectors ω_i). The matrix $T^{-1}FT$ is a diagonal matrix with diagonal elements equal to the eigenvalues of F, i.e.,

$$T^{-1}FT = \begin{bmatrix} \lambda_1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \lambda_2 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & \lambda_n \end{bmatrix} \tag{8}$$

where T is the modal matrix of F.

To transform the matrix F to the companion form, the matrix T (which is not unique (Ref. 16)) may be selected as

$$T = \begin{bmatrix} h \\ hF \\ \cdot \\ \cdot \\ hF^{n-1} \end{bmatrix} \tag{9}$$

where

$$h(1 \times n) = [1, 0, \ldots\ldots, 0]. \tag{10}$$

The matrix $T^{-1}FT$ is then the companion form given by

$$T^{-1}FT = \begin{bmatrix} 0 & 1 & 0 & \cdot & \cdot & 0 \\ 0 & 0 & 1 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & -\frac{a_2}{a_n} & \cdot & \cdot & -\frac{a_{n-1}}{a_n} \end{bmatrix} \tag{11}$$

where a_i, i = 0, 1.., n are the coefficients of the characteristic polynomial of F. That is,

$$a_n \lambda^n + a_{n-1} \lambda^{n-1} \ldots\ldots + a_0 = |F - \lambda I|. \tag{12}$$

Inspection of the standard, Jordan canonical, and companion forms (Eqs. (3), (8) and (11), respectively) and the filter equation (Eq. (5)) indicates that the required number of arithmetic operations (additions and multiplications) is a function of model representation. The model representation selected on the basis of minimum number of arithmetic operations would be the Jordan canonical form. However, the Ref. 18 study shows that accuracy of digital filters varies with filter structure even though the overall transfer functions of the digital filters are identical. This variance in accuracy is due to the finite word lengths associated with digital electronic implementation. Therefore, this study investigated the impact of filter structure on system performance. The approach adopted for determining the appropriate structure is discussed in the next section.

To code the controller on a digital computer the filter equations for state estimation (Eq. (5)) are generally represented by (1) state prediction equations and (2) state update equations. Filter equations -- obtained by expanding closed-loop filter dynamics in a Taylor series -- which predict state variables at time $(t + \Delta t)$, given measurements to time t, are described by

$$\hat{w}(t+\Delta t/t) = (I+\phi)\hat{w}(t/t)$$

$$\phi = T^{-1}FT\,\Delta t + \frac{(T^{-1}FT)^2\Delta t^2}{2!} + \frac{(T^{-1}FT)^3\Delta t^3}{3!} + \cdots . \tag{13}$$

When the sampling interval has been chosen, ϕ may be computed off-line to the desired accuracy. The number of terms retained in the series expansion for ϕ depends upon the microprocessor word length to be used in implementing the controller. That is, the number of terms retained is selected so that the sum of terms beyond the last term retained is less than the accuracy of the microprocessor. The notation $f(t + \Delta t/t)$ represents the value of the function f at time $(t + \Delta t)$ given measurements to time t. The filter update and deterministic control equations are given by

$$\hat{w}(t+\Delta t/t+\Delta t) = \hat{w}(t+\Delta t/t) + T^{-1}H\Delta t\left[z(t+\Delta t) - ET\hat{w}(t+\Delta t/t)\right]$$
$$u(t+\Delta t/t+\Delta t) = GT\hat{w}(t+\Delta t/t+\Delta t) . \tag{14}$$

Equation (13) indicates that state prediction computational requirements depend upon the system model employed in the filter and the state order; whereas, Eq. (14) shows that state update and input computational requirements depend on state and measurement orders and input and state orders, respectively. In addition, Eqs. (13) and (14) show that the arithmetic operations for solving the filter and deterministic control equations are matrix multiplications and additions.

For digital implementation system dynamics are normalized so that the control law (Eqs. (13) and (14)) may be coded using fixed-point rather than floating-point arithmetic. Fixed-point arithmetic is employed for digital implementation of the control logic to increase computational speed. Generally, system dynamics are normalized so that the digital numbers in the control computations range between -1.0 and +1.0. Then the most significant bit (bit 0) in the computer word is the sign bit and the least significant bit (bit b-1; where b denotes the number of bits in the microprocessor word) represents $2^{(-b+1)}$. For notational convenience, (i.e., to avoid introducing new symbols for normalized system variables) Eqs. (1) through (14) will be used to represent normalized system dynamics in the remainder of this report.

Accuracy of the microprocessor estimation and control logic will depend upon (1) word length of the microprocessor and (2) word length of the input/output (I/O) devices. Due to finite microprocessor and I/O word lengths (where I/O devices include direct digital output effectors such as transducers with frequency

output and stepper motors as well as A/D and D/A converters) quantization errors occur in the digital representation of a number. For the scaling described previously, the absolute value of the maximum quantization error e is given by

$$e = 2^{(-b+1)}. \tag{15}$$

MICROPROCESSOR REQUIREMENTS

Key issues associated with microprocessor implementation of linear quadratic Gaussian (LQG) control logic are addressed in this section. An analytical procedure to establish microprocessor accuracy requirements is described first. Computational requirements including arithmetic as well as interface speed are then set forth. Memory requirements as a function of system order and system model within the filter are then presented.

Accuracy Requirements

Controller accuracy required to obtain satisfactory system performance will depend upon (1) the structure of the system model employed within the Kalman filter and (2) system sensitivity to controller errors. To determine system time response sensitivity to microprocessor word length (that is, to determine microprocessor accuracy requirements) a performance index

$$J = \int_0^\infty [(\text{SYSTEM RESPONSE})^* - (\text{SYSTEM RESPONSE})^\dagger]^2 \, dt \tag{16}$$

--where $(\ )^*$ represents the system response with the controller coded on a 36-bit floating-point computer (a very accurate controller) and $(\ )^\dagger$ represents the system response with the controller coded using b bits (with $b \leq 36$) and fixed-point arithmetic--was defined to provide a quantitative measure of performance degradation. Its computed value provides a direct indication of the transient performance degradation associated with microprocessor quantization errors.

The integrand of J was selected as quadratic in system input and output variables so that the value of J can be analytically computed from the known problem matrices. That is, the performance index

$$J = \int_0^\infty [(y^* - y^\dagger)' \, Q(y^* - y^\dagger) + (u^* - u^\dagger)' \, R(u^* - u^\dagger)] dt \tag{17}$$

where
- y^* = output response vector with 36-bit controller
- $y^\dagger$ = output response vector with b-bit controller
- u^* = control vector with 36-bit controller
- $u^\dagger$ = control vector with b-bit controller
- Q, R = weighting matrices

was defined. The performance index J represents performance degradation due to finite word lengths less than the accurate 36-bit word length. As the number of bits in the computer word approaches 36, J approaches zero.

From Eqs. (1) and (5) the closed-loop system dynamics (open-loop system and controller) are described by

$$\begin{aligned} \dot{\tilde{x}} &= \tilde{F}\ \tilde{x}\ ; \qquad \tilde{x}(0) = \tilde{x}_0 \\ y &= \tilde{C}\ \tilde{x} \\ u &= \tilde{G}\ \tilde{x} \\ \tilde{x} &= \begin{bmatrix} x \\ \hat{w} \end{bmatrix} \end{aligned} \tag{18}$$

where

$$\tilde{F} = \left[\begin{array}{c|c} A & B(GT)_c \\ \hline (T^{-1}H)_c E & (T^{-1}FT)_c - (T^{-1}H)_c\ (ET)_c \end{array}\right] \tag{19}$$

$$\tilde{C} = \left[\begin{array}{c|c} C & D(GT)_c \end{array}\right]$$

$$\tilde{G} = \left[\begin{array}{c|c} 0 & (GT)_c \end{array}\right] .$$

The notation $(\ )_c$ indicates that the matrix is implemented in the digital controller. The accuracy of these matrices in the controller is limited by the word length of the digital controller. Matrices without the subscript c represent the open-loop system dynamics. The solution to Eq. (18) is given by

$$\begin{aligned} \tilde{x} &= e^{\tilde{F}t}\ \tilde{x}_0 \\ y &= \tilde{C}\, e^{\tilde{F}t}\ \tilde{x}_0 \\ u &= \tilde{G}\, e^{\tilde{F}t}\ \tilde{x}_0 \ . \end{aligned} \tag{20}$$

Substituting Eq. (20) for the 36-bit representation (y^*, u^*) and for the b-bit representation ($y^\dagger$, $u^\dagger$) into Eq. (17) and algebraically manipulating the resulting equation for J as well as taking the expected value of J over x_0 leads to

$$J = \mathrm{tr}\ P_1 + 2\,\mathrm{tr}\ P_2 + \mathrm{tr}\ P_3 \tag{21}$$

where

$$\begin{aligned} \tilde{F}^{*\prime} P_1 + P_1 \tilde{F}^* &= -(\tilde{C}^{*\prime} Q \tilde{C}^* + \tilde{G}^{*\prime} R \tilde{G}^*) \\ \tilde{F}^{*\prime} P_2 + P_2 \tilde{F}^\dagger &= (\tilde{C}^{*\prime} Q \tilde{C}^\dagger + \tilde{G}^{*\prime} R \tilde{G}^\dagger) \\ \tilde{F}^{\dagger\prime} P_3 + P_3 \tilde{F}^\dagger &= -(\tilde{C}^{\dagger\prime} Q \tilde{C}^\dagger + \tilde{G}^{\dagger\prime} R \tilde{G}^\dagger) \end{aligned} \tag{22}$$

and "tr" indicates the trace of a square matrix. Equations (21) and (22) indicate that the performance index J is a function of (1) the closed-loop system dynamics ($\tilde{F}$, $\tilde{C}$, $\tilde{G}$) and (2) the microprocessor word length (number of bits). For a given closed-loop system the performance degradation J averaged over all possible initial

conditions can be computed by solving the bilinear equations (Eq. (22)) for P_1, P_2, and P_3. This analytical procedure eliminates the need to run system time responses and perform numerical integrations over many initial conditions to determine microprocessor accuracy requirements for implementing the digital controller. In addition, this procedure for establishing microprocessor accuracy requirements is not limited to LQG controllers. The procedure is applicable to any linear closed-loop system expressed in Eq. (18) form.

In addition to using Eq. (22) to establish the required microprocessor word length, Eq. (22) can also be employed to determine the structure (i.e., $\bar{F}$ -- see Eq. (19)) to minimize performance degradation. That is, time response y^*, u^* associated with the high accuracy 36-bit floating-point controller should have negligible dependence upon system structure since the system structures $\bar{F}$ are mathematically equivalent. On the other hand, time response $y^\dagger$, $u^\dagger$ will depend upon the particular structure selected because of fixed-point arithmetic operations and small word length (e.g., b = 4 or 8). Results obtained will assist in establishing the optimum structure for microprocessor implementation of LQG control logic.

Computational Requirements

In this section computational requirements are discussed. Computational requirements depend upon (1) the maximum sample time (i.e., minimum sampling rate) that maintains desired system performance and system stability as well as (2) arithmetic and I/O computation time per sampling interval. A procedure for determining the maximum sample time (i.e., maximum sampling interval) is described first. The maximum sample time depends upon system dynamics. Computation time per sampling interval is then presented. The computation time per sampling interval depends upon (1) the number of arithmetic and I/O operations per sampling interval, and (2) the speed of the microprocessor and interface. Computation time per sampling interval establishes the minimum sample time. If required computation time per sampling interval is less than the maximum sample time that maintains desired system response, then the microprocessor and interface system may be used to implement the LQG controller.

Sample Time

To analyze linear, time-invariant, discrete-time systems (Eqs. (13) and (14)), the z-transform is employed. From Eqs. (13) and (14), discrete controller dynamics are given by

$$\hat{w}(k+1) = \phi_D \hat{w}(k) + H_D z(k+1)$$
$$u(k+1) = G_D \hat{w}(k+1) \tag{23}$$

where

$$\phi_D = (I - T^{-1}HET\Delta t)(I + \phi)$$
$$\phi = T^{-1}FT\Delta t + \frac{(T^{-1}FT)^2\Delta t^2}{2!} + \frac{(T^{-1}FT)^3\Delta t^3}{3!} + \cdots\cdots \tag{24}$$
$$H_D = T^{-1}H\Delta t$$
$$G_D = GT$$

and k denotes the k^{th} sample time. Routh's criterion, the root-locus method, or frequency response methods -- procedures used to investigate transient response and stability of continuous-time systems -- may be extended to investigate transient response and stability of discrete-time systems (Ref. 19).

In this study, the root-locus method was employed to relate sample time (Δt) and system performance. This method has the advantage that the z-plane poles are available on the root-locus plot once the controller has been selected and the root locus plotted as a function of sample time. The effect on system performance of different sample times is then easily determined from the root-locus plot.

To establish transient response based on system poles in the z-plane, the relationships

$$z = e^{\sigma\Delta t} e^{j\omega\Delta t}$$
$$s = \sigma + j\omega \tag{25}$$

are employed. Equation (25) shows that

$$|z| = e^{\sigma\Delta t}$$
$$z = e^{j\omega\Delta t} \tag{26}$$

Therefore, for a given Δt, (1) lines of constant σ map into circles and (2) lines of constant ω map into rays originating at the origin. To establish lines of constant damping (ξ) in the z-plane the relationships

$$\sigma = \xi\omega_n$$
$$\omega = \omega_n\sqrt{1-\xi^2} \tag{27}$$

along with Eq. (26) are used. Lines of constant damping (for ξ = 0 to 1.0) as well as lines of constant $\sigma\Delta t$ and $\omega\Delta t$ are shown in Fig. 3.

The system will be stable for all values of $\sigma\Delta t$ for which the root locus is inside the unit circle ($\sigma\Delta t < 0$). For transient response, Fig. 3 shows that:

(1) for a pole on the positive real axis ($\omega = 0$) and inside the unit circle the time response is exponentially decaying. The rate of decay increases as the pole approaches the origin. There is no decay if the pole is on the unit circle.

(2) for complex poles in the right-half plane and inside the unit circle the time response is an exponentially decaying sinusoidal oscillation. The frequency of oscillation increases as the poles move toward the imaginary axis. The rate of decay is faster near the origin. Stable oscillations result if the poles are on the unit circle.

(3) for a pole on the negative real axis ($\omega\Delta t = \pi$) as well as for complex poles in the left-half plane (and inside the unit circle) the time response is an exponentially decaying sinusoidal oscillation. The frequency of oscillation is higher than the right-half plane frequencies. Decay is faster near the origin.

Transient response for a discrete-time system as a function of z-plane pole location is summarized in Fig. 4.

The z-transform for the discrete controller to be implemented on microprocessors (Eq. (23)) is given by

$$u(z) = K(z)\, z(z) \tag{28}$$

where

$$K(z) = G_D (Iz - \phi_D)^{-1} H_D z.$$

Note that the transfer matrix K (z) is a function of sample time (see Eqs. (23) and (24)). To relate sample time (Δt) and system performance a root-locus as a function of sample time is plotted. That is, the poles of K (z) as a function of Δt are plotted in the z-plane. Accuracy of the gains in the transfer matrix K are limited by the computer word length established by applying the performance index discussed previously. From the root-locus plot the maximum sample time for which system performance is adequate may be determined.

By examining z-plane poles as a function of Δt the maximum sample time for stable controller operation (all poles within the unit circle) may be established. In addition, controller transient response as a function of sample time may be investigated (see Fig. 4). The maximum sample time for which all controller real poles are in the right-half plane (i.e., high oscillations associated with real poles in the left-half plane are not present in the controller time response) may be determined. Based on this analysis the control designer may determine the maximum sample time required for adequate controller response. This sample time must be smaller than the closed-loop system sample time based upon the Nyquist criteria. Also, the sample time selected must be long enough to complete all arithmetic and I/O computations.

Computation Time

The computation time per sampling interval includes (1) arithmetic computation time and (2) I/O computation time. Arithmetic computation time depends upon (1) the number of arithmetic operations per sampling interval and (2) the microprocessor time to execute each operation. The I/O computation time depends upon (1) the number of inputs and outputs and (2) the speed of the interfaces. In this study, it is assumed that the interfaces are A/D and D/A converters.

To reduce the number of computational operations, Eq. (23) rather than Eqs. (13) and (14) are coded on the microprocessor. Note that the matrix T should be selected so that ϕ_D is the desired structure.

The software to implement Eq. (23) on a microprocessor consists of matrix/vector multiplications and one vector addition. For computational speed the matrix/vector multiplications ($p\ (q \times 1) = M(q \times r)\ v\ (r \times 1)$) are accomplished by (1) multiplying all q elements in the j^{th} column of the matrix M by v_j (i.e., $P_{ij} = M_{ij}\ v_j$ for i = 1 to q) for j = 1 to r and (2) adding the r columns of the i^{th} row of P to obtain p_i for i = 1 to q. This procedure for matrix/vector multiplication is more efficient than directly coding $p_i = \sum_{j=1} M_{ij}\ v_j$ because v_j only has to be addressed once rather than r times for the matrix/vector multiplication.

A block diagram of the matrix/vector multiplication code is shown in Fig. 5. Figure 5 indicates that, in addition to the time for multiplying or adding the appropriate numbers, there is logic time associated with each multiplication and addition. The multiplication, multiplication logic, addition, and addition logic times are denoted by $\bar{M}$, $\bar{L}_{m1}$, $\bar{L}_{m2}$, $\bar{A}$, $\bar{L}_{a1}$, and $\bar{L}_{a2}$, respectively. These times are a function of microprocessor.

The arithmetic computations per sampling interval for the three structures -- standard, Jordan canonical, and companion forms -- are compared in Table I. The structure employed within the filter affects only the filter equation. Table I indicates that for n > 1 the Jordan canonical form requires the minimum arithmetic computations per sampling interval. Also, the effect of structure on computational requirements is most significant when the number of states is significantly greater than the number of outputs and controls since differences in the filter equation computational requirements are a function only of the number of states (n).

The computation time as a function of addition, multiplication, logic, and A/D and D/A conversion times as well as state, input, and output orders is shown in Table I. The arithmetic and I/O operation times vary with microprocessor and A/D and D/A converters, respectively. The characteristics of a representative set of microprocessors and A/D and D/A converters are shown in the Appendix (Tables A-I and A-II, respectively). The microprocessor characteristics include word length, internal registers, indexed addressing capabilities, and multiply instruction capability. The A/D and D/A converter characteristics include word length, conversion time, and technology.

To compute the arithmetic operation time, code must be written to execute the matrix/vector multiplication (see Fig. 5). This code depends upon the microprocessor. Two microprocessors were selected to (1) demonstrate the implementation of LQG control logic on microprocessors and (2) define the computation time as a function of system order only. An Intel 8080 and a Digital Equipment Corporation (DEC) LSI 11/2 -- two very different microprocessors -- were chosen. Both processors are commercially available and extensively used. The Intel 8080 is an 8-bit microprocessor with a limited instruction set and no indexed addressing; whereas, the DEC LSI 11/2 is a 16-bit microprocessor with a powerful instruction set and indexed addressing. Table II lists representative microprocessor instructions for the Intel 8080 and DEC LSI 11/2. Note that hardware multiply/divide instructions are available for the DEC LSI 11/2.

The codes to implement LQG control logic on an Intel 8080 and DEC LSI 11/2 have been included in the Appendix, Figs. A1 and A2, respectively. These codes are for the controller with the standard structure in the filter. They would have to be modified to take advantage of reductions in computational requirements due to a Jordan canonical or companion structure in the filter. These codes use a software multiply subroutine. If a hardware multiply were available, the hardware multiply would replace the call to the software multiply subroutine. Characteristics of hardware multipliers are summarized in the Appendix (Table A-III). Hardware multiplier characteristics include word length, multiply time, and technology.

Table III shows times associated with executing the matrix/vector multiplication on an Intel 8080 and DEC LSI 11/2. The add and multiply logic times as well as the add time are significantly less for the DEC LSI 11/2 than for the Intel 8080. This decrease in execution time with an increase in word length is due primarily to the more powerful instruction set and addressing modes of the DEC LSI 11/2. On the other hand, the software multiply time for the DEC LSI 11/2 is 4.25 times the execution time for the Intel 8080. This increase is due to the fact that the word length of the DEC LSI 11/2 (16 bits) is double the word length of the Intel 8080 (8 bits). For comparison, the execution time for a double precision (16 bit) software multiply on the Intel 8080 is shown in Table III. This time is greater than the software multiply time on the DEC LSI 11/2. Hardware and software multiply instruction times for the DEC LSI 11/2 are also compared in Table III. This comparison indicates that the hardware multiply instruction time on the DEC LSI 11/2 is approximately 3.5 percent of the software multiply time.

Intel 8080 and DEC LSI 11/2 arithmetic computation times with software multiply subroutines are compared in Table IV. Table IV shows that the computation time per sampling interval for the DEC LSI 11/2 is significantly greater than the computation time per sampling interval on the Intel 8080 due to the greater DEC LSI 11/2 multiply time. Table IV also compares DEC LSI 11/2 software multiply arithmetic computation time with DEC LSI 11/2 hardware multiply arithmetic computation time. This comparison indicates that the arithmetic computation time per sampling interval on the DEC LSI 11/2 can be decreased by approximately 90 percent by using a hardware multiply. Table IV also shows that computation time is more influenced by the number of states (n) than by the number of inputs (m) or outputs (ℓ).

Computation time per sampling interval as a function of microprocessor, system structure within the controller, and system order is shown in Fig. 6. It is assumed that the system has the same number of states, inputs, and outputs. Figure 6 shows that for a given sampling interval the number of states that can be in the controller varies with microprocessor. For example, for a sampling interval of 25 msec the maximum number of states in a standard structure controller is: 2 states for the DEC LSI 11/2 with software multiply, 4 states for the Intel 8080 with software multiply, and 8 states for the DEC LSI 11/2 with hardware multiply. Figure 6 also shows that the computation time with the Jordan canonical structure within the controller is significantly less than the computation time with the standard structure within the controller.

The arithmetic and I/O computation requirements per sampling interval (Table I) establish the minimum sampling interval. If this minimum sampling interval is less than the sample time required to achieve adequate system performance, then the microprocessor/interface system may be used to implement the LQG controller.

Memory Requirements

Memory requirements depend upon (1) the system model and (2) the computer code including temporary storage to implement the LQG control algorithm. System model memory requirements are a function of model structure as well as system state, input, and output orders. System model requirements will not vary with microprocessor. On the other hand, the computer code and temporary storage requirements will vary with microprocessor as well as system model.

Memory requirements as a function of system order and structure are shown in Table V for the Intel 8080 and DEC LSI 11/2. Table V indicates that the system and temporary storage memory requirements are primarily dependent on the number of states in the system. The standard structure requirements increase with the square of the number of states; whereas, the Jordan canonical and companion structures increase linearly with the number of states. Table V shows that memory requirements for the Jordan canonical and companion structures are the same. The Intel 8080 computer code memory requirements (341 words) are approximately triple the DEC LSI 11/2 computer code memory requirements (113 words). This difference in code requirements is due to the fact that the DEC LSI 11/2 instruction set is more powerful than the Intel 8080 instruction set.

Figure 7 shows PROM and RAM memory requirements as a function of model structure within the controller for the Intel 8080 microprocessor. It is assumed in Fig. 7 that the number of system states, inputs, and outputs are equal. Figure 7 indicates that memory requirements for the LQG controller are not severe. For example, if the system has 10 states, 10 inputs, and 10 outputs then the PROM and RAM memory requirements are 0.641 K and 0.365 bytes, respectively. On the other hand, the arithmetic computation time per sample interval for this 10 state controller implemented on the DEC LSI 11/2 with hardware multiply (the fastest microprocessor considered) would be 33.5 msec. Therefore, computation requirements are a more critical consideration than memory requirements.

APPLICATION AND EVALUATION OF MICROPROCESSOR REQUIREMENT PROCEDURES

The application and evaluation of microprocessor requirement procedures are discussed in this section. To evaluate and illustrate the developed procedures they were applied to (1) a second-order plant and (2) a fifth-order F100 turbofan engine model linearized at sea level static military operation. To verify the results a continuous system/microprocessor controller was simulated on the UNIVAC 1110 at UTRC.

The closed-loop simulation for verifying the procedures is described first. Use of these procedures to define microprocessor requirements for digital control of a second-order system are presented next. In the final section results of applying the procedures to establish microprocessor requirements of a linear quadratic Gaussian (LQG) controller for the fifth-order F100 turbofan model are set forth.

Closed-Loop Simulation for Verifying Procedures

The closed-loop simulation to be used in verifying the procedures for establishing microprocessor requirements to implement LQG control logic is shown in Fig. 8. The simulation consists of (1) the open-loop system dynamics (Eq. (1)) and (2) discrete controller dynamics (Eq. (23)). The open-loop system was simulated using floating-point arithmetic and a very small integration step size (0.001 seconds) so that the plant appears as a continuous system. The discrete control and estimation logic were coded to simulate fixed-point arithmetic operation. Coding was developed in the controller so that the word length and sample time can be varied. The word length can be varied from 1 to 36 bits. The resulting closed-loop system appears as a continuous plant/discrete controller.

The analysis for establishing the microprocessor requirement procedures is based on linear system theory. However, when the LQG control algorithm is coded on a digital computer the algorithm is nonlinear due to finite word lengths. That is, the matrix multiply

$$p_1 = m_{11} v_1 + m_{12} v_2 \tag{29}$$

is calculated in the computer as

$$(p_l)_c = \left[(m_{11c} v_{1c})_c + (m_{12c} v_{2c})_c\right]_c. \tag{30}$$

However, in the analysis it is assumed that

$$(p_l)_c = (m_{11c} v_{1c} + m_{12c} v_{2c})_c. \tag{31}$$

This assumption in the analysis (Eq. (31)) must be taken into account when interpreting results for very small word lengths (i.e., the filter is very nonlinear - (Eq. 30)).

Second-Order System

Normalized plant, control, and filter dynamics (Eqs. (1) and (2)) are shown in Table VI. In control design the deterministic control weighting matrices were selected as identity matrices.

The performance index for the second-order plant as a function of word length and model structure within the controller is shown in Fig. 9. For notational convenience the performance index is normalized so that the performance index scale ranges from 0 to 1.0. Figure 9 indicates that closed-loop system performance improves significantly (J decreases rapidly) as the controller word length increases from 3 to 8 bits. If less than 3 bits are employed in the controller the closed-loop system is unstable. As the controller word length is increased from 8 to 16 bits small change in closed-loop system performance occurs. In addition, Fig. 9 indicates that the model structure employed within the filter has little effect on the closed-loop system sensitivity to controller word length for word lengths greater than or equal to seven bits. However, for word lengths less than seven bits and for this closed-loop system the Jordan canonical form is slightly less sensitive than the companion and standard forms to controller errors due to finite microprocessor word length.

Output response of the second-order model with different word-length controllers is compared in Fig. 10. The standard structure was employed in the controller. Output response with different structures and 6 bit word length in the controller are shown in Fig. 11. The controller sample time in Figs. 10 and 11 is 0.1 seconds. The response in Figs. 10 and 11 results from initial conditions of 0.5 on the normalized states. Transient response of Figs. 10 and 11

verify results obtained from analytically computing the performance index J. In addition, second-order system time responses with different sample times in the controller were generated. Different controller sample times did not change the relative effect of different controller word lengths on system performance. Note that the steady-state bias errors shown in Figs. 10 and 11 for controller word lengths less than or equal to 8 bits do not affect the performance index (Eq. (17)). (If they did, the value of the performance index would approach infinity.) This result is due to the fact that the performance index is computed from Eq. (18); whereas, response of the continuous system/discrete controller simulation is computed from Eqs. (1), (2), and (23).

To determine the required sample time for the second-order 8 bit controller a root locus of the controller poles as a function of sample time is plotted. Recall that the requirement for 8-bit accuracy is established using the performance index. The root locus plot is shown in Fig. 12. Poles at selected sample times (0.1, 0.7 and 1.3 seconds) are displayed. Figure 12 indicates that for sample times less than or equal to 0.7 seconds the poles are in the right-half plane and are heavily damped (see Fig. 3). Therefore, for sample times less than or equal to 0.7 seconds the time response of the controller should not be oscillatory. In addition, response decay time will decrease as the sample time increases from 0.1 to 0.7 seconds. Figure 12 shows that one of the poles is close to the unit circle in the left-half plane when the sample time is equal to 1.3 seconds. Therefore, for a sample time of 1.3 seconds the controller response will be stable but oscillatory. The maximum frequency of the closed-loop system is 0.54 hertz. To satisfy the Nyquist rate the sample time must be less than 0.93 seconds. Therefore, the sample time of 0.7 seconds satisfies the Nyquist rate. The maximum sample time for the second-order controller is 0.7 seconds. With this sample rate the system response will not be oscillatory.

Second-order output response as a function of sample time is shown in Fig. 13. The standard structure with 8 bit accuracy was employed within the controller. The output response results from an initial condition of 0.5 on both normalized states. Figure 13 verifies results obtained from the z-plane plot analysis.

Computation times required per sampling interval for the second-order system using software multiplication are listed in Table VII. This table indicates that the minimum computation time is 2.73 msec for the Jordan canonical structure coded on the Intel 8080. The maximum computation time is 10.12 msec for the standard structure coded on the DEC LSI 11/2. Both times are well within the maximum required sample time of 700 msec.

Memory requirements for the second-order controller are shown in Table VIII. Table VIII shows that model structure within the second-order controller has little effect on memory requirements due to the low system order.

In summary, results indicate that the Intel 8080 with software multiply can be used to implement the LQG control law for this second-order system. The requirements for the second-order controller are (1) 8-bit word length accuracy, (2) a minimum sample time (based on computations) of 2.73 msec and a maximum sample time (based on controller poles) of 700 msec, and (3) 347 words of PROM and 21 words of RAM.

Fifth-Order Gas Turbine Engine Model

Closed-loop engine dynamics are shown in Table IX. The plant is a fifth-order F100 turbofan engine model linearized at sea level static military operation. The deterministic control design matrices were identity matrices. The process and measurement noise statistics are shown in Table IX.

The normalized performance index for the fifth-order engine model as a function of word length is displayed in Fig. 14. The standard model structure was employed within the controller for the results illustrated in Fig. 14. Transforming the closed-loop standard engine model to the companion or Jordan canonical structures resulted in a numerically ill-conditioned closed-loop matrix. That is, the magnitudes of the elements in the closed-loop matrix varied over a wide range. For example, the magnitude of the companion form closed-loop matrix elements varied from 0.076 to 5.357×10^9. As a result of this numerical variation the closed-loop system dynamics were unstable for word lengths less than or equal to 16 bits. Figure 14 shows that closed-loop engine performance with the standard model structure within the controller improves as the word length increases from 7 to 12 bits. Increasing the word length from 12 to 16 bits does not significantly affect closed-loop performance.

Afterburner pressure response of the engine model -- representative of closed-loop engine response -- with different word-length controllers is compared in Fig. 15. The responses in Fig. 15 result from initial conditions of 0.1 on the normalized engine states. Figure 15 verifies results obtained from analytically computing the performance index J. However, the technique to analytically and in closed form evaluate the performance degradation due to a finite word length in the controller (Eqs. (21) and (22)) eliminates the need to perform these numerical integrations.

To establish the maximum sample time the poles of the 12-bit discrete controller as a function of sample time are plotted in the z-plane. Poles at selected sample times (0.010, 0.025 and 0.046 seconds) are displayed in

Fig. 16. Figure 16 indicates that for sample times less than or equal to 0.025 seconds the poles of the discrete controller are in the right-half plane. Response for sample times less than or equal to 0.025 seconds will be non-oscillatory. Figure 16 shows that for a sample time of 0.046 seconds controller response will be oscillatory. For sample times greater than 0.047 seconds there is a pole outside the unit circle (that is, the controller is unstable). The maximum frequency of the closed-loop system is 5.4 hertz. Therefore, the Nyquist sample time is 0.09 seconds. The maximum sample time of 0.025 seconds is less than the Nyquist sample time. Therefore, the arithmetic and I/O computation time per sampling interval must be less than or equal to 25 msec to achieve satisfactory performance. Figure 17 verifies results of the z-plane plot analysis.

Computation times required per sampling interval for the fifth-order engine model are listed in Table X. The times shown are for the standard structure within the controller. Recall that controllers with the companion or Jordan canonical structure were not stable for word lengths of 16 bits or less. Also, only computation times for the DEC LSI 11/2 microprocessor are shown since a word length of 12 bits is the accuracy required to implement the controller. Table X indicates that a sampling interval time of 87.48 msec is required to complete the control arithmetic and I/O computations if a software multiply subroutine is used. Since the maximum sample time from the z-plane plot analysis is 25 msec, the DEC LSI 11/2 with software multiply is not fast enough for implementing the controller. On the other hand, the total computation time for the DEC LSI 11/2 with a hardware multiply is 9.68 msec. This time is less than the required maximum sample time of 25 msec. Therefore, the DEC LSI 11/2 with the hardware multiply package can be used to implement this LQG F100 engine controller.

Memory requirements (see Table XI) for the linearized F100 engine controller are less than 200 words of RAM and 200 words of PROM. These memory requirements are very small.

In summary, the DEC LSI 11/2 with a hardware multiply can be used to implement the LQG F100 engine controller. The requirements for the linearized F100 engine model LQG controller are (1) 12-bit word length accuracy, (2) a minimum sample time (based on computations) of 9.68 msec and a maximum sample time (based on controller poles) of 25.0 msec, and (3) less than 200 words of RAM and 200 words of PROM.

CURRENT UTRC PROGRAM

The current UTRC program sponsored by AFOSR is a logical extension of the research reported here. The program is directed toward (1) validating results presented in this report by demonstrating microprocessor implementation of LQG control logic and (2) extending the analysis of microprocessor requirements to nonlinear systems.

To demonstrate microprocessor implementation of LQG control logic, a system that consists of an analog computer, a microcomputer, A/D and D/A converters, and a display unit is being employed. Continuous linear system dynamics are simulated on the analog computer. Discrete LQG control logic is coded on the microcomputer. General purpose A/D and D/A converters are employed to interface the microcomputer controller with the analog computer simulation.

To extend the analysis to nonlinear systems, scheduled-gain nonlinear estimation and control logic based on LQG theory is being employed. Requirements -- including accuracy, computational capability, and memory requirements -- for implementing nonlinear estimation and control logic are being established. The procedures employed to establish microprocessor requirements for linear systems are being extended to determine microprocessor requirements for nonlinear systems.

REFERENCES

1. Michael, G. J. and G. S. Sogliero: Key Control Assessment for Linear Multivariable Systems. Proceedings, 15th IEEE Conference on Decision and Control, December 1-3, 1976, Clearwater Beach, Florida, pp. 1120-1125.

2. Farrar, F. A. and G. J. Michael: Estimation for Advanced Technology Engines. Proceedings, 17th IEEE Conference on Decision and Control, January 10-12, 1979, San Diego, California, pp.342-347.

3. Farrar, F. A. and G. J. Michael: Nonlinear Stochastic Control Design for Gas Turbine Engines. Office of Naval Research Report ONR-CR215-247-2F, Final Technical Report prepared under Department of the Navy Contract N00014-76-C-0710, June 1978.

4. Farrar, F. A. and G. J. Michael: Multivariable Control Synthesis for F100 Engine Stability Reset Operation. United Technologies Research Center Report UTRC77-23, Technical Report prepared under P&WA Government Products Division Engineering Order Supplement No. 802707, March 1977.

5. Michael, G. J. and F. A. Farrar: Estimation and Control of Critical Gas Turbine Variables. ASME Winter Annual Meeting, Paper WA/AUT-12, New York, New York, December 1976.

6. DeHoff, R. L. and W. E. Hall: Multivariable Control Design Principles with Application to the F100 Turbofan Engine. Proceedings, 1976 Joint Automatic Control Conference, Lafayette, Indiana, July 1976.

7. Michael, G. J. and F. A. Farrar: Development of Optimal Control Modes for Advanced Technology Propulsion Systems. United Technologies Research Center Report N911620-2, Annual Technical Report prepared under Department of the Navy Contract N00014-73-C-0281, March 1974.

8. Michael, G. J. and F. A. Farrar: An Analytical Method for the Synthesis of Nonlinear Multivariable Feedback Control. United Technologies Research Center Report M941338-2, Final Technical Report prepared under Department of the Navy Contract N00014-72-C-0414, June 1973.

9. Spang III, H. A.: The Challenge of Microprocessors. IEEE Control Systems Newsletter, October 1976.

REFERENCES
(Continued)

10. Barnich, R. G.: Designing Microcomputers into Your Products. Instrument Society of America ISA-76 International Conference and Exhibit, Houston, Texas, October 1976.

11. Grossman, R. M.: "Micro" Peripherals - Moving to Meet the Microcomputer Challenge. EDN, February 5, 1976, pp. 38-47.

12. Bishop, P. G.: Microprocessors: Computing in Miniature. Physics in Technology, March 1976, pp. 47-53.

13. Larson, A.: Some Questions and Answers about Microprocessors. Proceedings of the IFAC 6th World Congress, Boston/Cambridge, Massachusetts, August 1975.

14. Hopkins, H. G.: An 8 Bit Microprocessor Identifies Second Order Transfer Functions. Proceedings, 1976 IEEE Conference on Decision and Control, Clearwater, Florida, December 1976.

15. Farrar, F. A.: Microprocessor Implementation of Advanced Control Modes. Proceedings, 1977 Summer Computer Simulation Conference, Chicago, Illinois, July 1977, pp. 339-342.

16. Bryson, A. E., Jr. and Y. C. Ho: Applied Optimal Control. Ginn and Company, Waltham, Mass., 1969.

17. Luenberger, D. G.: Canonical Forms for Linear Multivariable Systems. IEEE Trans. Auto. Control, Vol. AC-12, No. 3, June 1967, pp. 290-293.

18. Gold, B. and C. M. Radar: Digital Processing of Signals. McGraw-Hill Book Company, 1969.

19. Gupta, S. C. and L. Hasdorff: Fundamentals of Automatic Control. John Wiley & Sons, Inc. 1970.

LIST OF SYMBOLS

A	Constant n x n matrix used to describe linear system dynamics
a_i	Coefficients of the characteristic polynomial of the matrix F
B	Constant n x m matrix used to describe linear system dynamics
b	Number of bits in digital word
C	Constant p x n matrix used to describe linear system dynamics
$\tilde{C}$	Constant p x 2n matrix used to describe closed-loop linear system dynamics
D	Constant p x m matrix used to describe linear system dynamics
E	Constant ℓ x n matrix used to describe linear system dynamics
E_T	Constant ℓ x n matrix used to describe transformed linear system dynamics
e	Quantization error
F	Constant n x n matrix used to describe optimal deterministic closed-loop system dynamics
$\tilde{F}$	Constant 2n x 2n matrix used to describe optimal stochastic closed-loop system dynamics
F_T	Constant n x n matrix used to describe transformed optimal closed-loop system dynamics
G	Constant m x n optimal deterministic closed-loop feedback gain matrix
$\tilde{G}$	Constant m x 2n optimal stochastic closed-loop feedback gain matrix
G_D	Constant m x n optimal deterministic closed-loop feedback gain matrix for microprocessor implementation
G_T	Constant m x n optimal closed-loop feedback gain matrix for transformed system

LIST OF SYMBOLS (Continued)

H	Constant n x ℓ Kalman filter gain matrix
H_D	Constant n x ℓ Kalman filter gain matrix for microprocessor implementation
H_T	Constant n x ℓ Kalman filter gain matrix for transformed system state model
h	1 x n row vector with $h_1 = 1.0$; $h_i = 0$, i = 2, ... n
I	Identity matrix
i	General subscript
J	Performance index
j	General subscript
K	Transfer matrix for discrete controller
k	Discrete time
ℓ	Dimension of system measurement vector z
m	Dimension of system input vector u
msec	Milliseconds
n	Dimension of system state vector x
nsec	Nanoseconds
P_i	Unknown matrices in the bilinear equations for computing the performance index J (i = 1,2,3)
p	Dimension of system output vector y
Q	Constant p x p output weighting matrix in quadratic performance index
R	Constant m x m input weighting matrix in the quadratic performance index

LIST OF SYMBOLS (Continued)

RAM	Read/Write Memory
PROM	Programmable Read Only Memory
s	Laplace transform operator
sec	Seconds
t	Time, sec
T	Constant nonsingular n x n state transformation matrix
u	m x 1 system input vector
u^*	m x 1 optimal system input vector
w	n x 1 transformed system state vector
x	n x 1 system state vector
$\tilde{x}$	2n x 1 closed-loop system state vector
$\tilde{x}_o$	Closed-loop system state vector at t = 0
y	p x 1 system output vector
z	ℓ x 1 system measurement vector
$\mathcal{Z}$	Discrete transform operator
η	ℓ x 1 sensor noise vector
λ_i	Eigenvalue of closed-loop system (i = 1, 2, ... n)
μsec	Microseconds
ξ	m x 1 input (process) noise vector
σ	Real part of complex frequency variable s
ϕ	$T^{-1}\,FT\,\Delta t + \frac{(T^{-1}FT)^2\,\Delta t^2}{2!} + \ldots\ldots$

LIST OF SYMBOLS (Continued)

ϕ_D	Constant n x n closed-loop system matrix for microprocessor implementation
ω	Imaginary part of complex frequency variable s
ω_i	n x 1 eigenvector associated with the ith eigenvalue
$( )_j$	jth component of vector in parentheses
$( )_{ij}$	Element in the ith row, jth column of the matrix in parentheses
$( )^*$	System response or controller matrix associated with 36-bit controller
$( )^\dagger$	System response or controller matrix associated with b-bit controller ($b \leq 36$ bits)
$( )_c$	Matrix in parentheses implemented on digital controller
$(\dot{})$	Time derivative of quantity in parentheses
$d( )$	Differential of quantity in parentheses
$\Delta( )$	Finite increment of quantity in parentheses
$(\hat{})$	Estimated value of quantity in parentheses
$( )^{-1}$	Inverse of matrix in parentheses
$( )'$	Transpose of quantity in parentheses
tr( )	Trace operator--trace of square matrix is equal to sum of all diagonal elements of the matrix
$\triangleq$	Equals by definition

TABLE I

COMPUTATION TIME

Function \ Structure		Standard	Jordan Canonical	Companion
Addition	Filter	$n(n-1) + n(\ell-1) + n$	$n(\ell-1) + n$	$(n-1) + n(\ell-1) + n$
	Control	$m(n-1)$	$m(n-1)$	$m(n-1)$
Multiplication	Filter	$n^2 + n\ell$	$n + n\ell$	$n + n\ell$
	Control	nm	nm	nm
Interface	Input	m	m	m
	Output	ℓ	ℓ	ℓ
Time *(μsec)		$(\bar{A}+\bar{La}1)(n^2+n\ell+mn-n-m)$ $+\bar{La}2(2n+m)$ $+(\bar{M}+\bar{Lm}1)(n^2+n\ell+nm)$ $+\bar{Lm}2(2n+\ell)$ $+\ell(A/D) + m(D/A)$	$(\bar{A}+\bar{La}1)(n\ell+mn-m)$ $+\bar{La}2(n+m)$ $+(\bar{M}+\bar{Lm}1)(n+n\ell+nm)$ $+\bar{Lm}2(n+\ell)$ $+\ell(A/D) + m(D/A)$	$(\bar{A}+\bar{La}1)(n+n\ell+mn-m-1)$ $+\bar{La}2(n+m)$ $+(\bar{M}+\bar{Lm}1)(n+n\ell+nm)$ $+Lm2(n+\ell)$ $+\ell(A/D) + m(D/A)$

*See Fig. 5 for definition of $\bar{A}$, $\bar{La}1$, $\bar{La}2$, $\bar{M}$, $\bar{Lm}1$, $\bar{Lm}2$
A/D denotes time for A/D conversion (μsec)
D/A denotes time for D/A conversion (μsec)

TABLE II

REPRESENTATIVE MICROPROCESSOR INSTRUCTIONS

R = Register
(R) = Contents of Register
RP = Register Pair
PC = Program Counter
INTEL 8080: Accumulator is Register When No Register Specified
M Indicates H, L Register Pair
DEC LSI 11/2: R6 is Stack Pointer

Instruction Description	Instruction		Length (Bytes)		Time (μsec)	
	Intel 8080	DEC LSI 11/2	Intel 8080	DEC LSI 11/2	Intel 8080	DEC LSI 11/2
Add Data From Memory to Register	ADD M	ADD(R), R	1	2	3.5	4.9
Call Subroutine Identified by Operand	CALL (SUB)	JSR PC,(SUB)	3	2	8.5	5.9
Double Precision Add	DAD RP	---	1	-	5.0	-
Decrement Register by One	DCR R	DEC, R	1	2	2.5	4.9
Input Data from Location Identified by Operand to Register	IN PORT	MOV LABEL, R	2	2	5.0	8.4
Transfer Program Control to Location Identified by Operand	JMP LABEL	BR LABEL	3	2	5.0	8.4
Load Immediate 16-Bit Data into Register	LXI RP, DATA	MOV DATA, R	3	4	5.0	4.9
Move Data from one Register to Another	MOV R,R	MOV R, R	1	2	2.5	3.5
Move Contents of Register to Memory	MOV M,R	MOV R,@#ADDR	1	4	3.5	6.6
Move Immediate Data to Register	MVI R, DATA	MOV R, #DATA	2	4	3.5	4.5
Move Immediate Data to Memory	MVI M, DATA	MOV#DATA,@#ADDR	2	6	5.0	8.0
Logical OR Register with Register	ORA R	XOR R,R	1	2	2.0	3.5
Output Data to Location Identified by Operand	OUT PORT	MOV R, LABEL	2	2	5.0	4.5
Place Top of Stack in Register	POP RP	MOV (R6)+, R	1	2	5.0	4.9
Place Register Contents on Top of Stack	PUSH RP	MOV R, -(R6)	1	2	5.5	5.2
Return from Subroutine	RET	RTS PC	1	2	5.0	5.2
One's Complement Register	CMA	COM R	1	2	2.0	4.2
Arithmetic Rotate Register Right/Left	---	ASR/L, R	-	2	-	5.8/3.8
Rotate Register Right/Left	RAR/L	ROR/L, R	1	2	2.0	5.2/3.8
Logical AND Register with Register	---	BIT R, R	-	2	-	3.5
Fixed-Point Multiply(1)	---	MUL R, R	-	2	-	37.0
Fixed-Point Divide(1)	---	DIV R, R	-	2	-	78.0
Floating-Point Add(1)	---	FADD R, R	-	2	-	42.1
Floating-Point Multiply(1)	---	FMUL R, R	-	2	-	121.1
Floating-Point Divide(1)	---	FDIV R, R	-	2	-	232.0

(1) Extended Arithmetic Chip, KEV11, Required

TABLE III

REPRESENTATIVE FIXED-POINT EXECUTION TIMES

Microprocessor		Time (μsec)						
		Add Logic		Add Instruction ($\bar{A}$)	Multiply Logic		Multiply Subroutine ($\bar{M}$)	Hardware Multiply Instruction
		$\bar{L}_{a1}$	$\bar{L}_{a2}$		$\bar{L}_{m1}$	$\bar{L}_{m2}$		
Intel 8080	8-Bit	13.5	76.0	20.5	81.0	60.5	271.0	-
	16-Bit (Double Precision)	-	-	-	-	-	>1200.0	-
DEC LSI 11/2		0.0	29.0	14.7	51.5	32.2	1152.0	40.5

TABLE IV

COMPARISON OF INTEL 8080 ARITHMETIC COMPUTATION TIME WITH DEC LSI 11/2 ARITHMETIC COMPUTATION TIME

Microprocessor	Arithmetic Computation Time* (μsec)		
	Standard Structure	Jordan Canonical Structure	Companion Structure
Intel 8080	$386.0(n^2+n\ell+nm)$ $+239.0n + 60.5\ell$ $+42.0m$	$386.0(n\ell+nm)$ $+488.5n + 60.5\ell$ $+42.0m$	$386.0(n\ell+nm)$ $+522.5n + 60.5\ell$ $42.0m-34.0$
DEC LSI 11/2	$1218.2(n^2+n\ell+nm)$ $+107.7n + 32.2\ell$ $+14.3m$	$1218.2(n\ell+nm)$ $+1264.7n + 32.2\ell$ $+14.3m$	$1218.2(n\ell+nm)$ $+1279.4n + 32.2\ell$ $+14.3m-14.7$
DEC LSI 11/2 with hardware multiply instruction	$106.7(n^2+n\ell+nm)$ $+107.7n + 32.2\ell$ $+14.3m$	$106.7(n\ell+nm)$ $+153.2n + 32.2\ell$ $+14.3m$	$106.7(n\ell+nm)$ $+167.9n + 32.2\ell$ $+14.3m-14.7$

*Total computation time = arithmetic computation time + I/O computation time. Arithmetic computation times shown here are subject to $\pm$ 20 percent variation. Representative I/O computation times are shown in the Appendix (Table A-II).

TABLE V

MEMORY REQUIREMENTS

Variable		Memory Type	Memory (Words)*		
			Standard Structure	Jordan Canonical Structure	Companion Structure
Past state estimate ($\hat{w}(k)$)		RAM	n	n	n
Current state estimate ($\hat{w}(k+1)$)		RAM	n	n	n
Measurement ($z(k+1)$)		RAM	ℓ	ℓ	ℓ
Control ($u(k+1)$)		RAM	m	m	m
System matrix (ϕ_D)		PROM	n^2	n	n
Kalman gain matrix (H_D)		PROM	$n\ell$	$n\ell$	$n\ell$
Control gain matrix (G_D)		PROM	mn	mn	mn
Temporary storage	Intel 8080	RAM	$n^2+n\ell+nm+2n+5$	$3n+n\ell+nm+5$	$3n+n\ell+nm+5$
	DEC LSI 11/2	RAM	$n^2+n\ell+nm+2n+22$	$3n+n\ell+nm+22$	$3n+n\ell+nm+22$
Computer code	Intel 8080	PROM	341	341	341
	DEC LSI 11/2	PROM	113	113	113
Total	Intel 8080	RAM	$n^2+4n+(n+1)\ (\ell+m)+5$	$5n+(n+1)\ (\ell+m)+5$	$5n+(n+1)\ (\ell+m)+5$
		PROM	$n^2+n(\ell+m)+341$	$n(\ell+m+1)+341$	$n(\ell+m+1)+341$
	DEC LSI 11/2	RAM	$n^2+4n+(n+1)\ (\ell+m)+22$	$5n+(n+1)\ (\ell+m)+22$	$5n+(n+1)\ (\ell+m)+22$
		PROM	$n^2+n(\ell+m)+113$	$n(\ell+m+1)+113$	$n(\ell+m+1)+113$

*Intel 8080: 1 word = 8 bits
DEC LSI 11/2: 1 word = 16 bits

R79-944258-2

TABLE VI

SECOND-ORDER MODEL DYNAMICS

Matrix	Matrix Elements
A	-3.414 -0.014 0.014 -0.586
B	-0.357 0.357
C	1.0 1.0
D	0.
E	1.0 1.0
G	-0.037 -0.201
H	-0.114 1.240
Covariance of ξ	0.01
Covariance of η	0.01

TABLE VII

COMPUTATION TIME REQUIRED FOR SECOND-ORDER SYSTEM

Microprocessor	Arithmetic Computation Time (msec)			I/O Computation Time (msec)*
	Standard Structure	Jordan Canonical Structure	Companion Structure	
Intel 8080	3.67	2.62	2.66	0.11
DEC LSI 11/2	10.01	7.45	7.46	0.11

*8-BIT Conversion times used (See Appendix - Table A-II):
A/D = 10μsec
D/A = 100μsec

TABLE VIII

MEMORY REQUIREMENTS FOR SECOND-ORDER SYSTEM

Variable		Memory Type	Memory (Words)*		
			Standard Structure	Jordan Canonical Structure	Companion Structure
Controller variables and temporary storage	Intel 8080	RAM	23	21	21
	DEC LSI 11/2	RAM	40	38	38
	Intel 8080	PROM	8	6	6
	DEC LSI 11/2	PROM	8	6	6
Computer code	Intel 8080	PROM	341	341	341
	DEC LSI 11/2	PROM	113	113	113
Total	Intel 8080	RAM	23	21	21
		PROM	349	347	347
	DEC LSI 11/2	RAM	40	38	38
		PROM	121	119	119

*Intel 8080: 1 word = 8 bits
DEC LSI 11/2: 1 word = 16 bits

TABLE IX

FIFTH-ORDER F100 ENGINE MODEL DYNAMICS

Engine Model Linearized at Sea-Level Static Military Operation

States	Outputs	Controls
Fan turbine inlet temperature	Airflow	Jet exhaust area
Main burner pressure	Fan stability margin	Fan inlet guide vanes
Fan speed	Compressor stability margin	Compressor variable vanes
Compressor speed	Thrust	Main burner fuel flow
Afterburner pressure	High Turbine inlet temperature	

Matrix	Matrix Elements				
A	-34.013	-9.303	12.037	-2.398	-1.254
	4.389	-38.762	-4.221	28.480	14.729
	-4.755	2.287	-0.400	-1.546	-2.200
	2.046	1.062	-0.729	-2.150	-0.624
	4.150	-8.814	-0.167	7.477	1.099
B	0.766	0.546	-0.813	17.095	
	0.056	1.341	7.737	8.641	
	0.156	-1.176	-0.416	2.034	
	-0.136	-0.024	-0.555	-0.378	
	-4.729	0.874	1.617	0.223	
C	-0.042	0.063	0.013	-0.054	1.404
	1.045	0.092	-0.060	-0.028	-0.050
	0.386	0.100	-0.217	0.170	-0.095
	0.305	-0.326	-0.458	0.584	-0.538
	-0.183	-0.564	0.394	-0.165	0.394
D	1.044	0.001	-0.013	0.002	
	-0.015	-0.003	-0.013	-0.044	
	-0.043	0.278	0.035	-0.155	
	-0.101	0.281	0.137	-0.041	
	0.073	0.047	-0.091	0.050	
E	1.0	0	0	0	0
	0	1.0	0	0	0
	0	0	1.0	0	0
	0	0	0	1.0	0
	0	0	0	0	1.0

(Continued)

TABLE IX (Continued)

Matrix	Matrix Elements				
G	1.602	-1.183	2.224	0.148	5.530
	0.012	3.074	-0.341	-0.903	-0.223
	-2.942	-5.064	5.544	-2.222	8.148
	-4.362	0.749	-0.652	-0.092	-0.811
H	6.110	1.879	20.560	2.456	0.036
	3.341	1.046	12.150	3.728	0.018
	0.822	0.273	3.493	2.523	0.006
	0.053	0.045	1.357	3.070	-0.002
	0.064	0.018	0.282	-.189	0.001
Covariance of ξ	0.01	0	0	0	0
	0	0.01	0	0	0
	0	0	0.01	0	0
	0	0	0	0.01	0
	0	0	0	0	0.01
Covariance of η	0.01	0	0	0	
	0	0.01	0	0	
	0	0	0.01	0	
	0	0	0	0.01	
	0	0	0	0	0.01

TABLE X

COMPUTATION TIME REQUIRED FOR F100 ENGINE MODEL

Standard Structure Within the Controller

Microprocessor	Computation Time	
	Arithmetic (msec)	I/O (msec)*
DEC LSI 11/2	86.03	1.45
DEC LSI 11/2 with hardware multiply instruction	8.23	1.45

*12-BIT Conversion times used (See Appendix - Table A-II):
A/D = 50μsec
D/A = 300μsec

TABLE XI

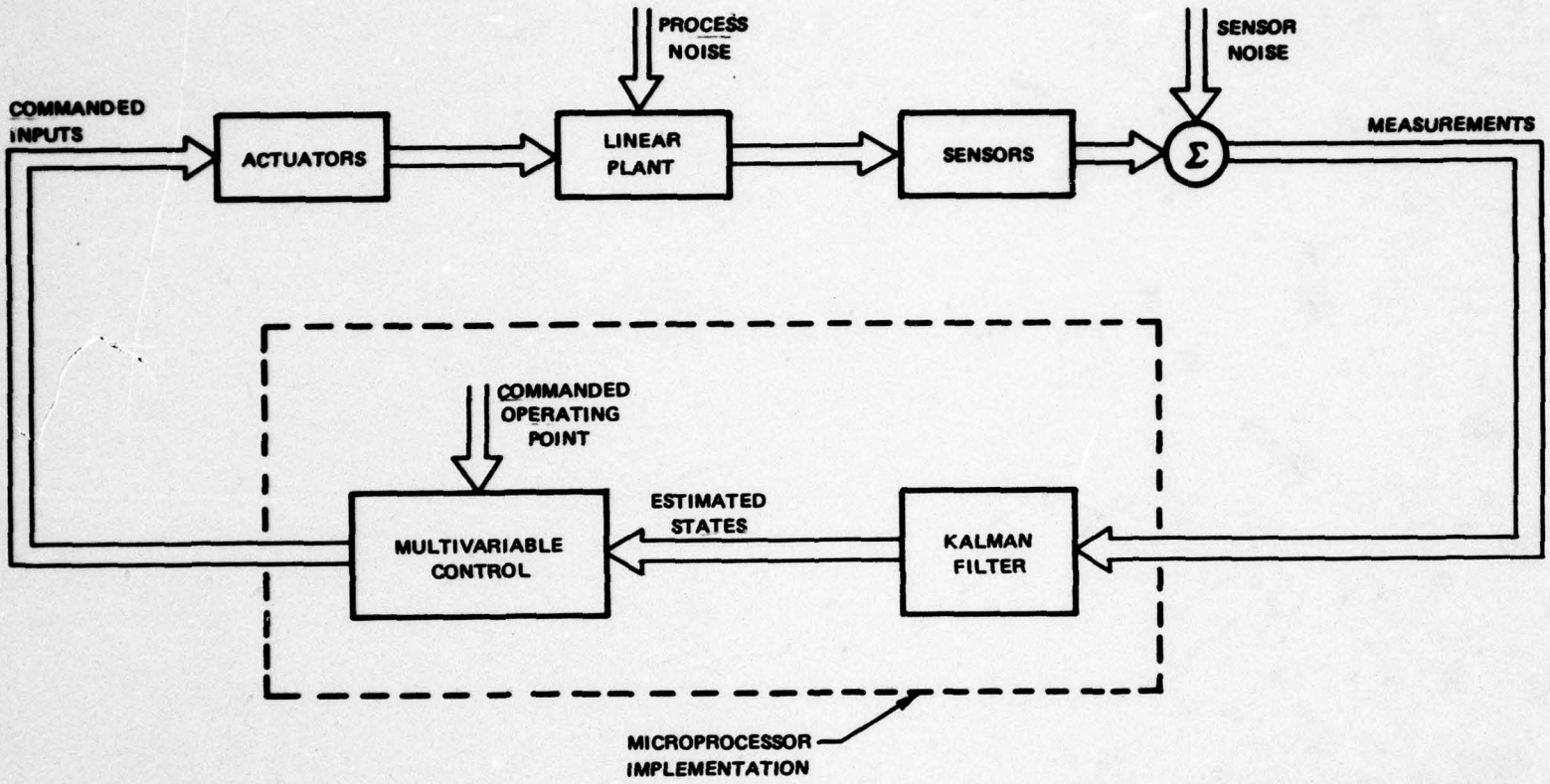
MEMORY REQUIREMENTS FOR FIFTH-ORDER ENGINE MODEL

Standard Structure Within Controller
DEC LSI 11/2 Microprocessor

Variable	Memory Type	Memory (words) (1 word = 16 bits)
Controller variables and temporary storage	RAM	121
	PROM	70
Computer code	PROM	113
Total	RAM	121
	PROM	183

R79-944258-2

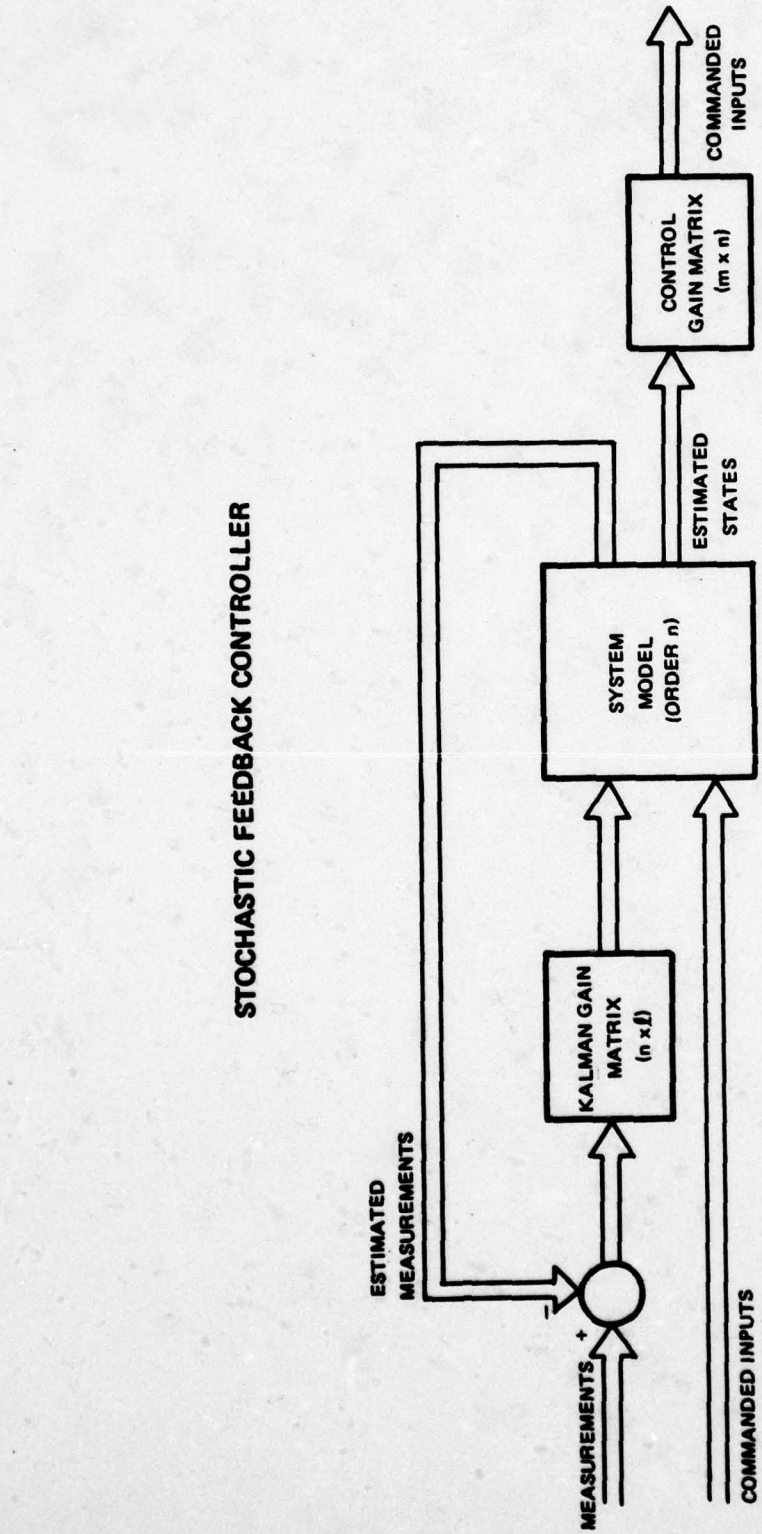
LINEAR STOCHASTIC FEEDBACK CONTROL STRUCTURE



77-06-243-2

FIG. 1

FIG. 2

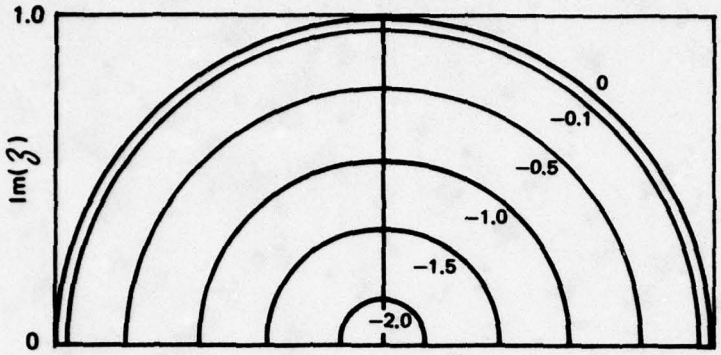


77-06-243-1

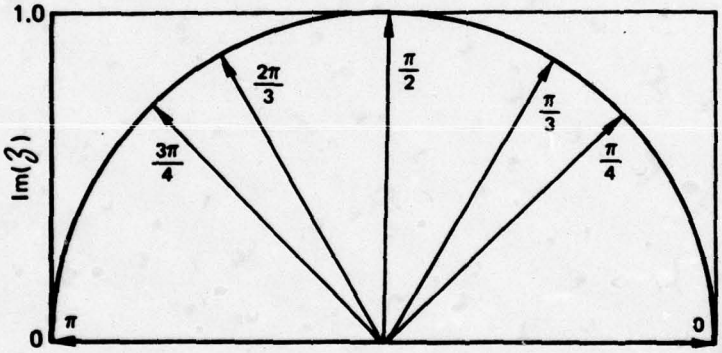
FIG. 3

TRANSFORMATION FROM S–PLANE TO $\mathcal{Z}$ –PLANE

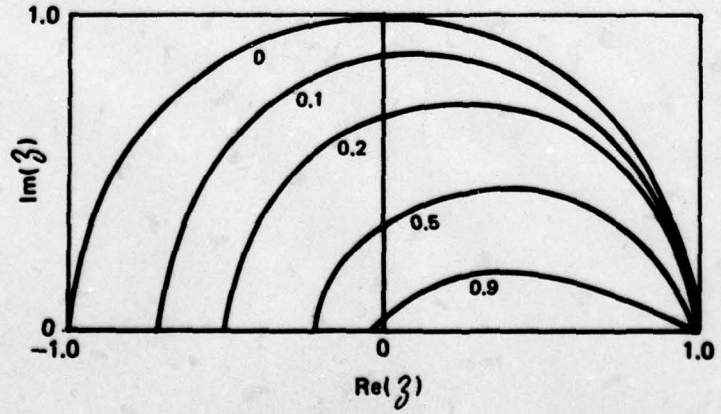
a) LINES OF CONSTANT $\sigma\Delta t$



b) LINES OF CONSTANT $\omega\Delta t$

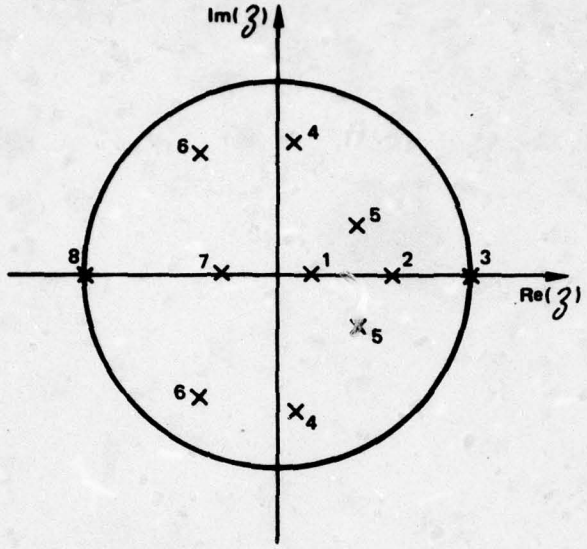


c) LINES OF CONSTANT DAMPING (ξ)



79–03–198–3

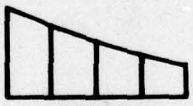
DISCRETE–TIME SYSTEM RESPONSE



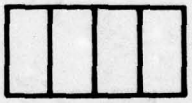
RESPONSE WITH POLES ON AXIS IN RIGHT HALF PLANE



POLE 1

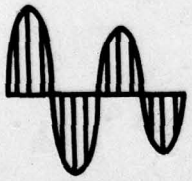


POLE 2



POLE 3

RESPONSE WITH COMPLEX POLES IN RIGHT HALF PLANE

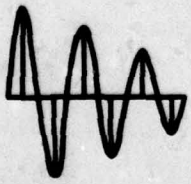


POLE 4

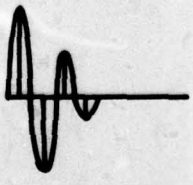


POLE 5

RESPONSE WITH POLES IN LEFT HALF PLANE



POLE 6



POLE 7



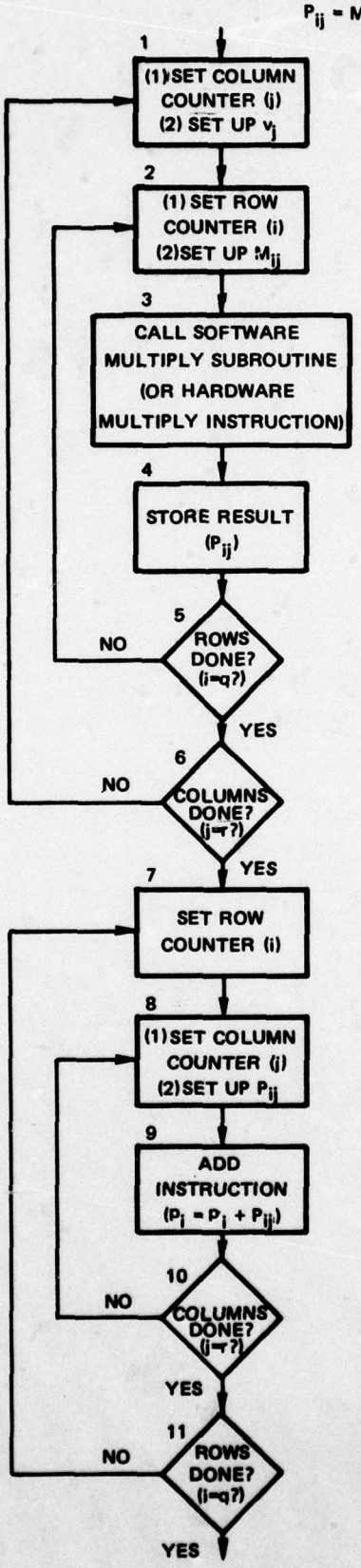
POLE 8

FIG. 5

BLOCK DIAGRAM FOR MATRIX/VECTOR MULTIPLICATION

$p(q \times 1) = M(q \times r)v(r \times 1)$

$P_{ij} = M_{ij}v_j$



$\overline{M}$ = TIME TO EXECUTE BLOCK 3
$\overline{\tau}_{m1}$ = TIME TO EXECUTE BLOCKS 2,4,5
$\overline{\tau}_{m2}$ = TIME TO EXECUTE BLOCKS 1,6
$\overline{A}$ = TIME TO EXECUTE BLOCK 9
$\overline{\tau}_{a1}$ = TIME TO EXECUTE BLOCKS 8,10
$\overline{\tau}_{a2}$ = TIME TO EXECUTE BLOCKS 7,11

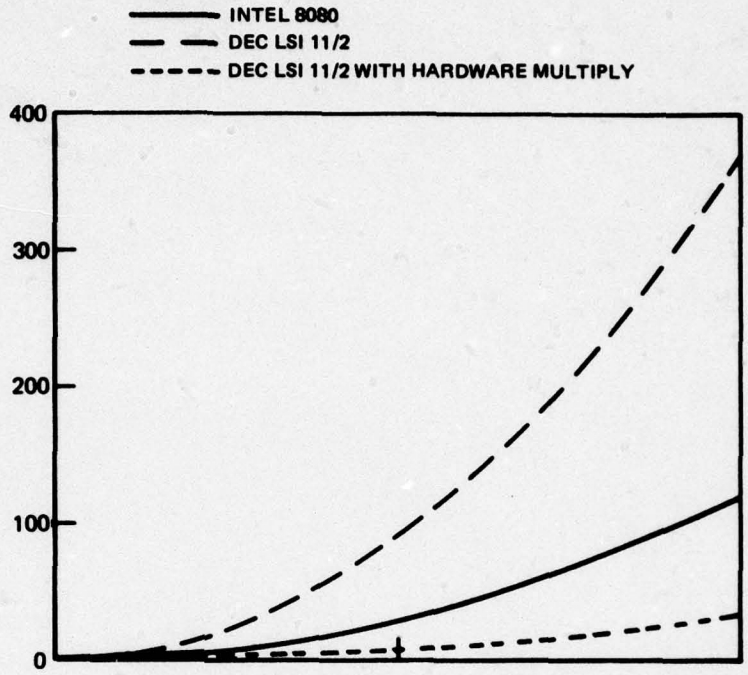
79–03–198–6

FIG. 6

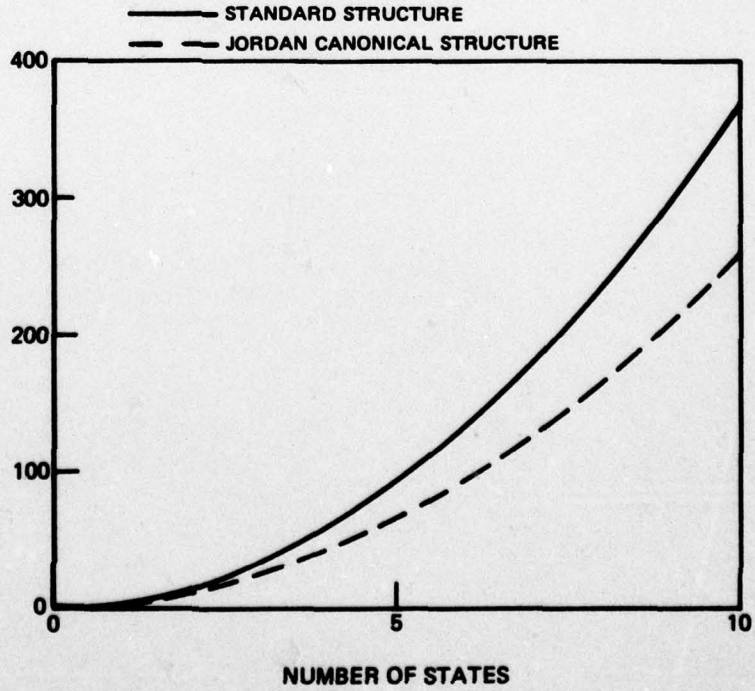
ARITHMETIC COMPUTATION TIME

$n = m = \ell$

a) COMPUTATION TIME AS A FUNCTION OF MICROPROCESSOR (msec); STANDARD STRUCTURE WITHIN CONTROLLER



b) COMPUTATION TIME AS A FUNCTION OF STRUCTURE WITHIN CONTROLLER (msec); DEC LSI 11/2 MICROPROCESSOR



79–03–198–5

FIG. 7

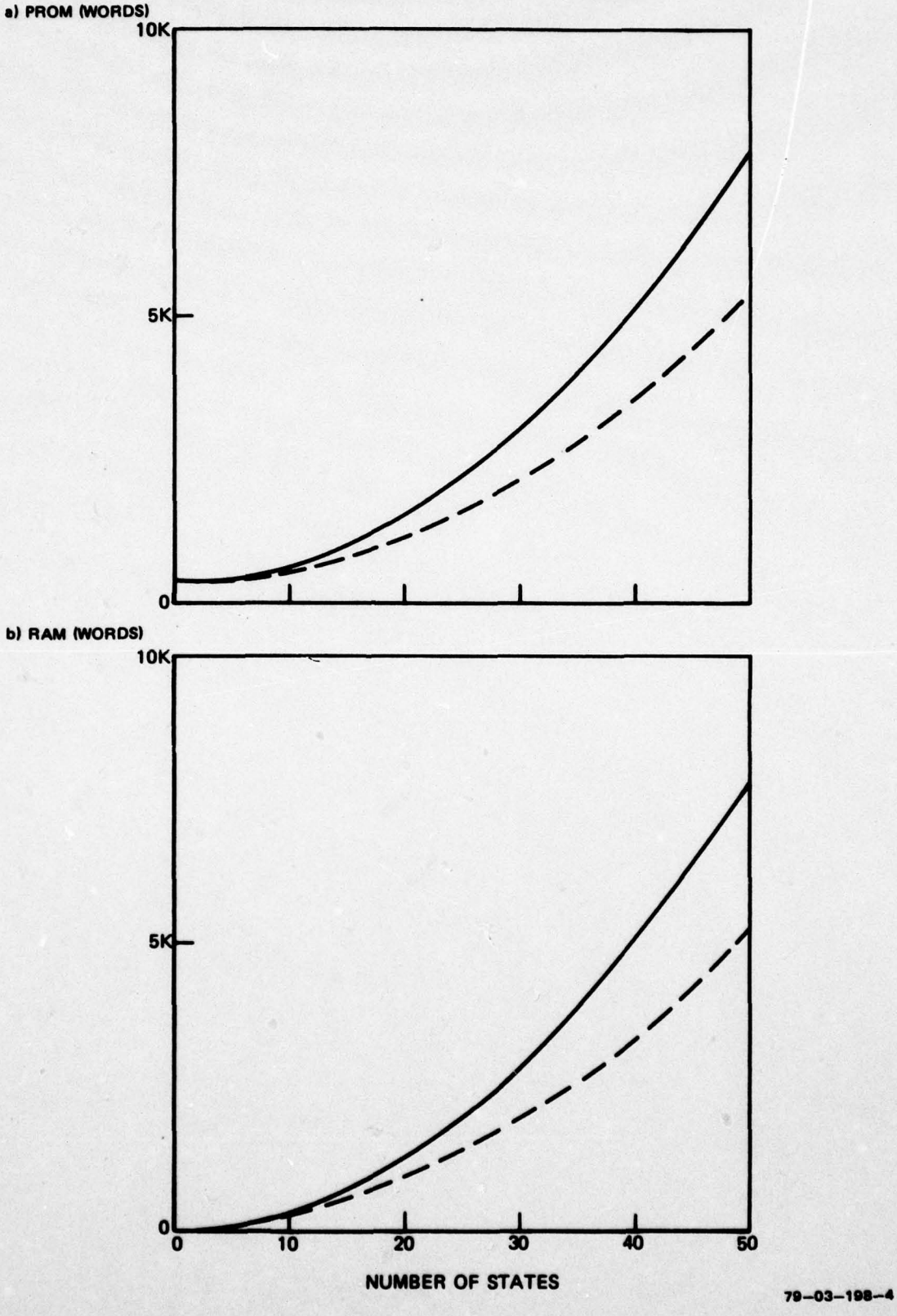
MEMORY REQUIREMENTS AS A FUNCTION OF STRUCTURE WITHIN CONTROLLER

INTEL 8080 MICROPROCESSOR

$n = m = \ell$

STANDARD STRUCTURE

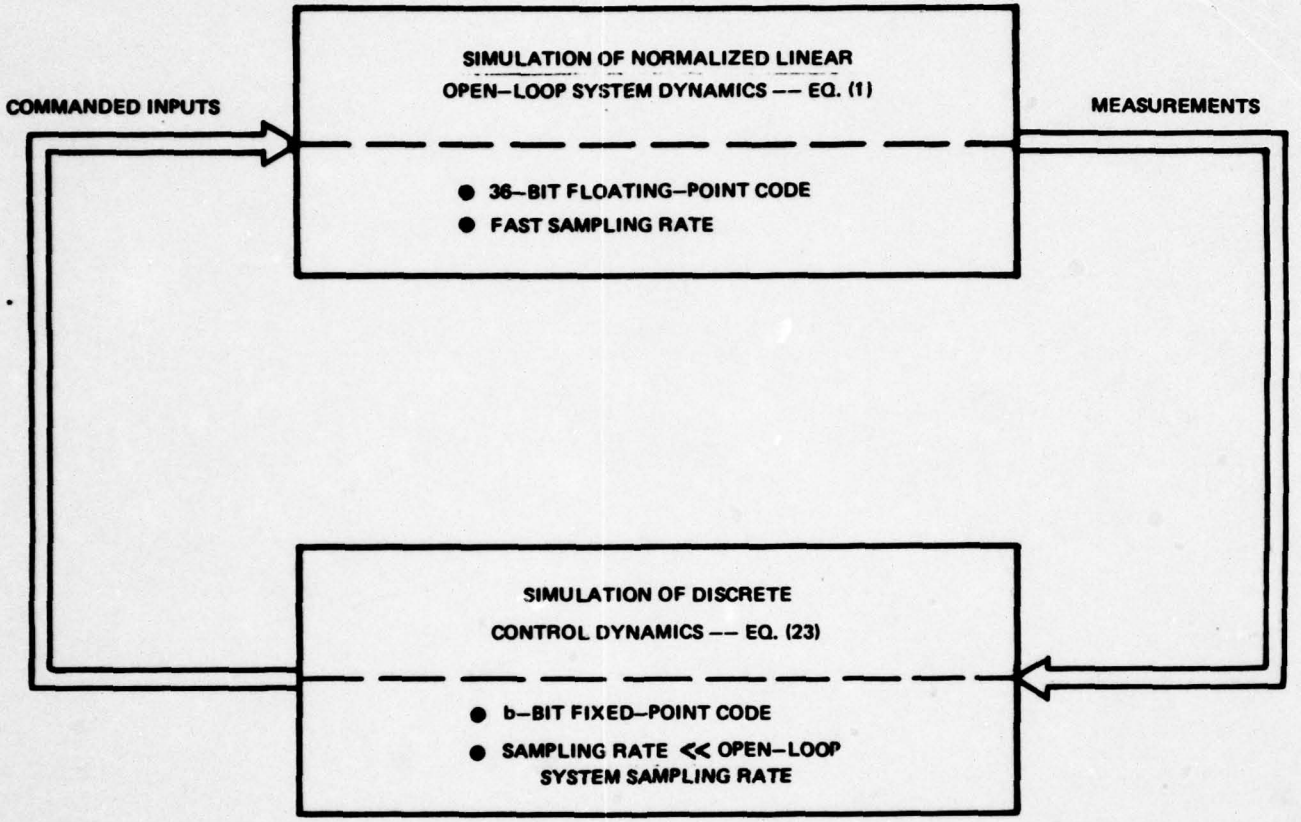
JORDAN CANONICAL OR COMPANION STRUCTURE



79–03–198–4

R79-944258-2

STOCHASTIC FEEDBACK CONTROL SIMULATION

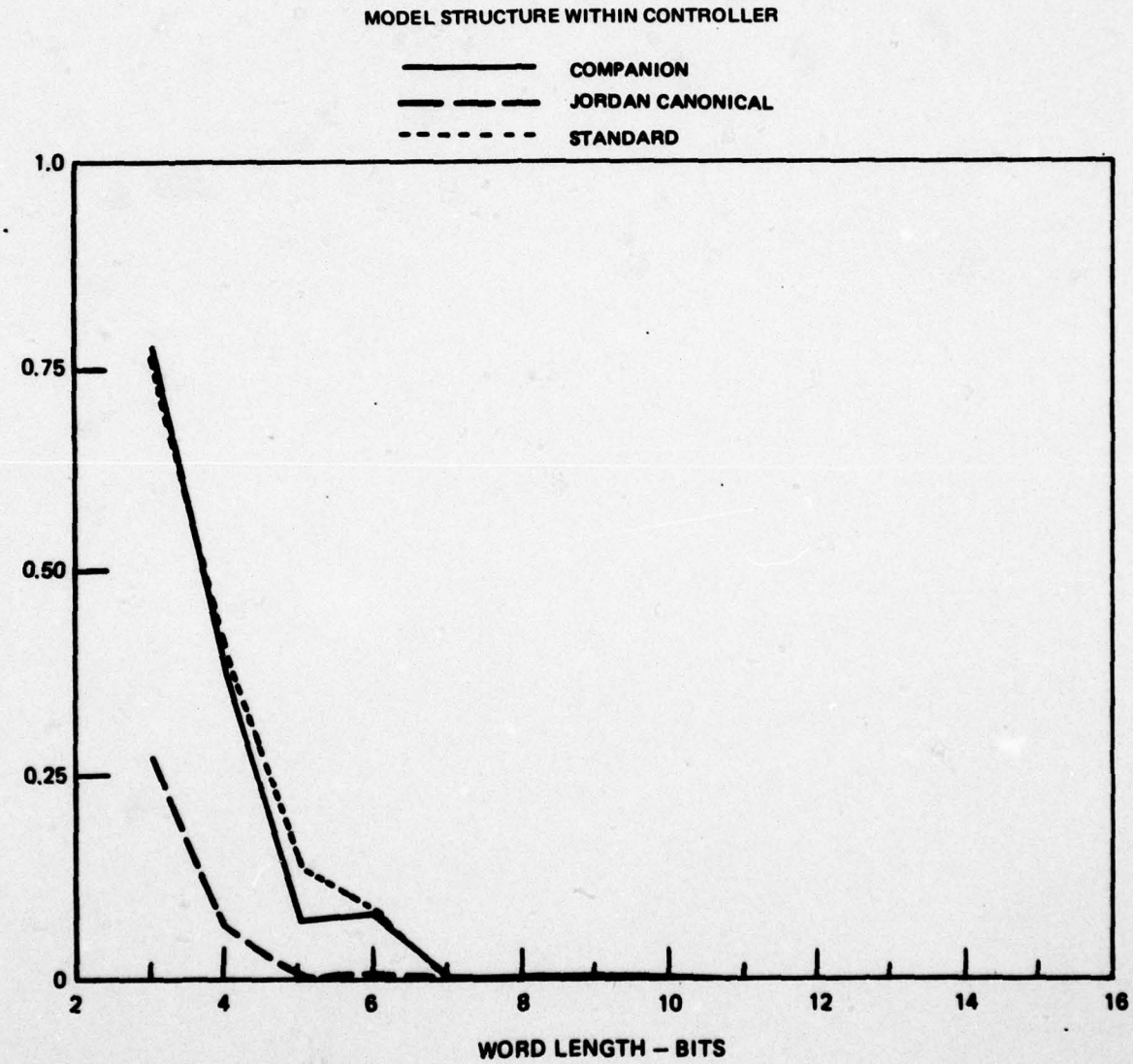


79–03–46–1

FIG. 8

FIG. 9

NORMALIZED PERFORMANCE INDEX FOR SECOND–ORDER SYSTEM



78–08–80–1

FIG. 10

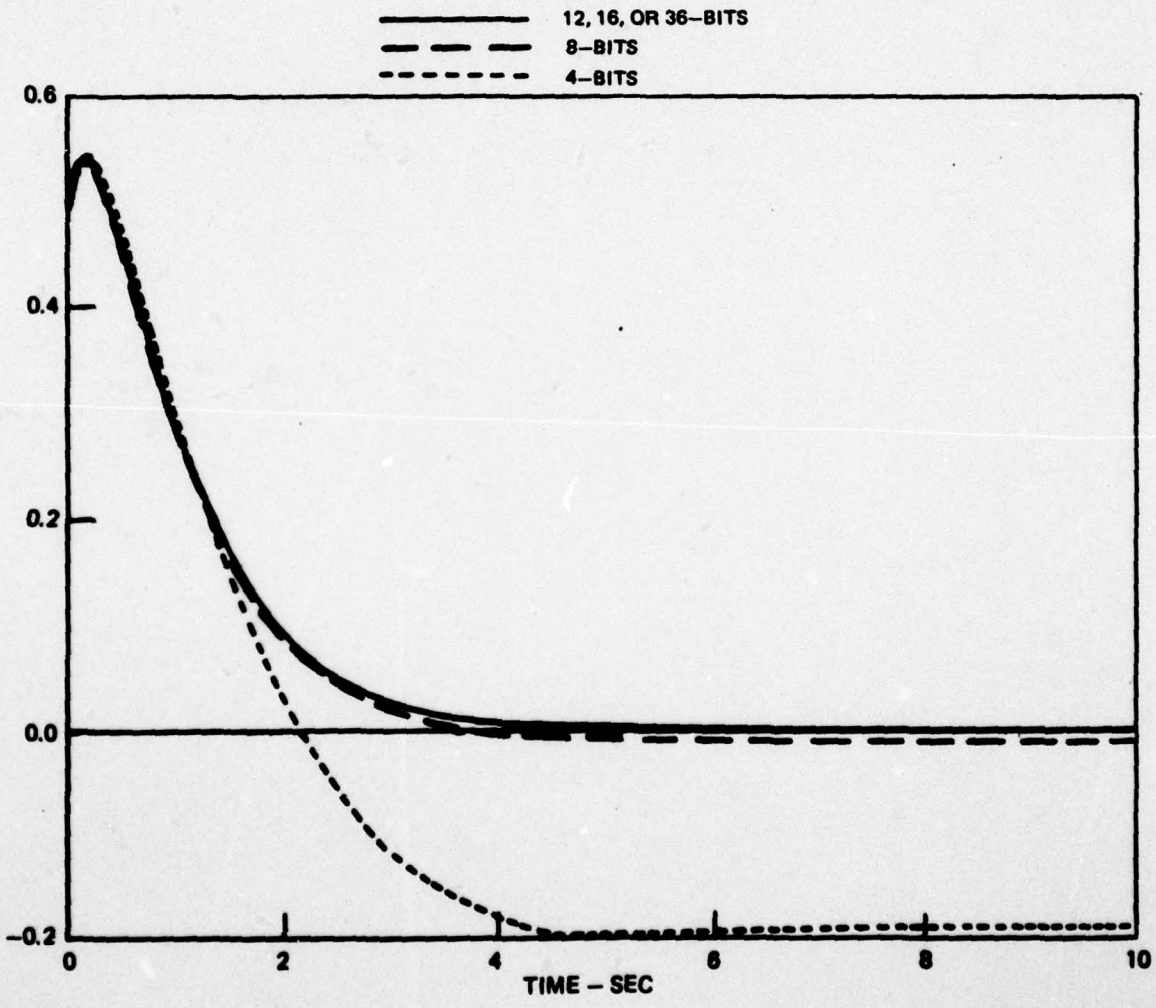
SECOND–ORDER OUTPUT RESPONSE AS A FUNCTION OF WORD LENGTH

RESPONSE TO INITIAL CONDITION: X'_0 = (0.5,0.5)

STANDARD STRUCTURE WITHIN CONTROLLER

Δt = 0.1 SEC

DIGITAL CONTROLLER WITH



78–08–80–3

FIG. 11

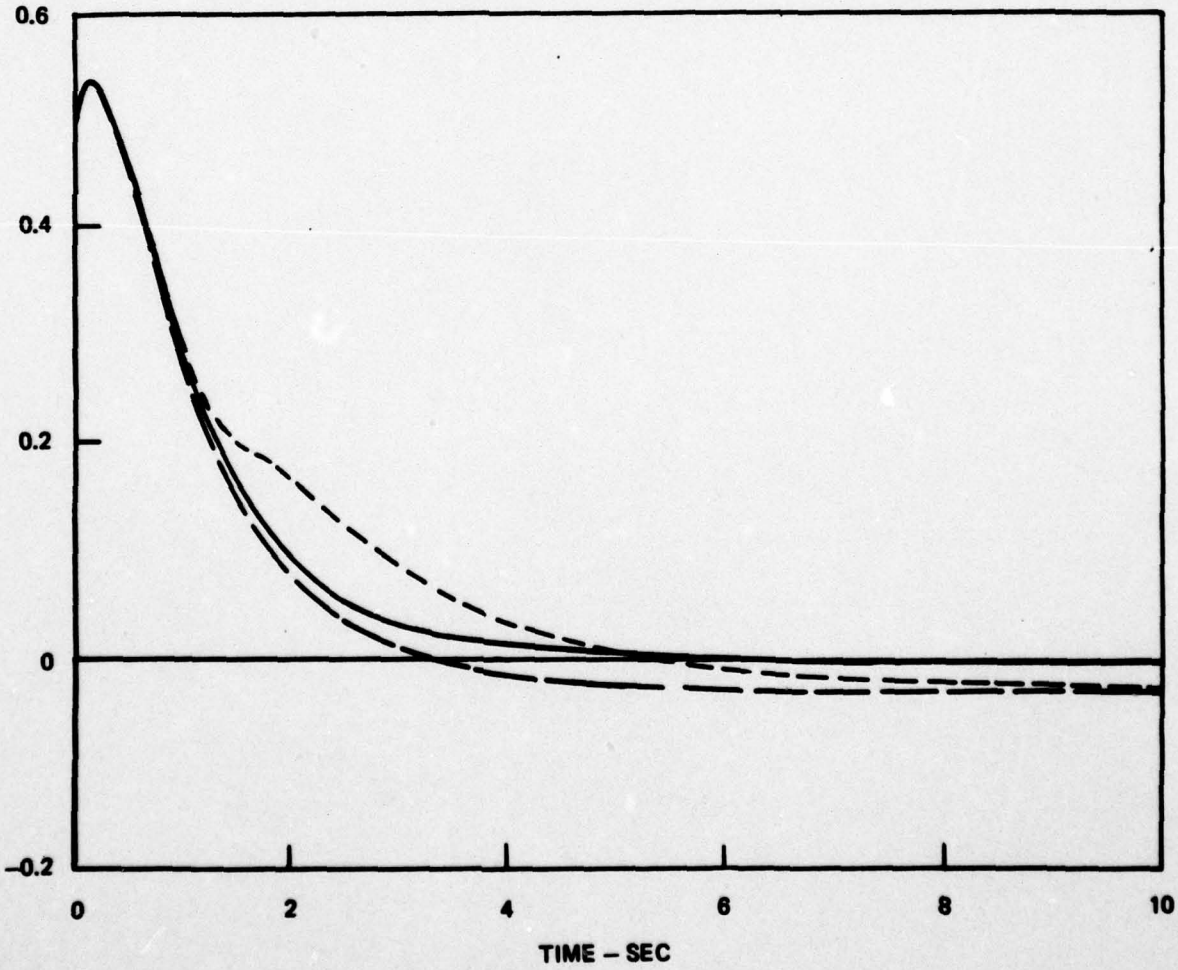
SECOND–ORDER OUTPUT RESPONSE AS A FUNCTION OF MODEL STRUCTURE WITHIN CONTROLLER

RESPONSE TO INITIAL CONDITION: $X_0' = (0.5, 0.5)$

$\Delta t = 0.1$ SEC

DIGITAL CONTROLLER WITH:

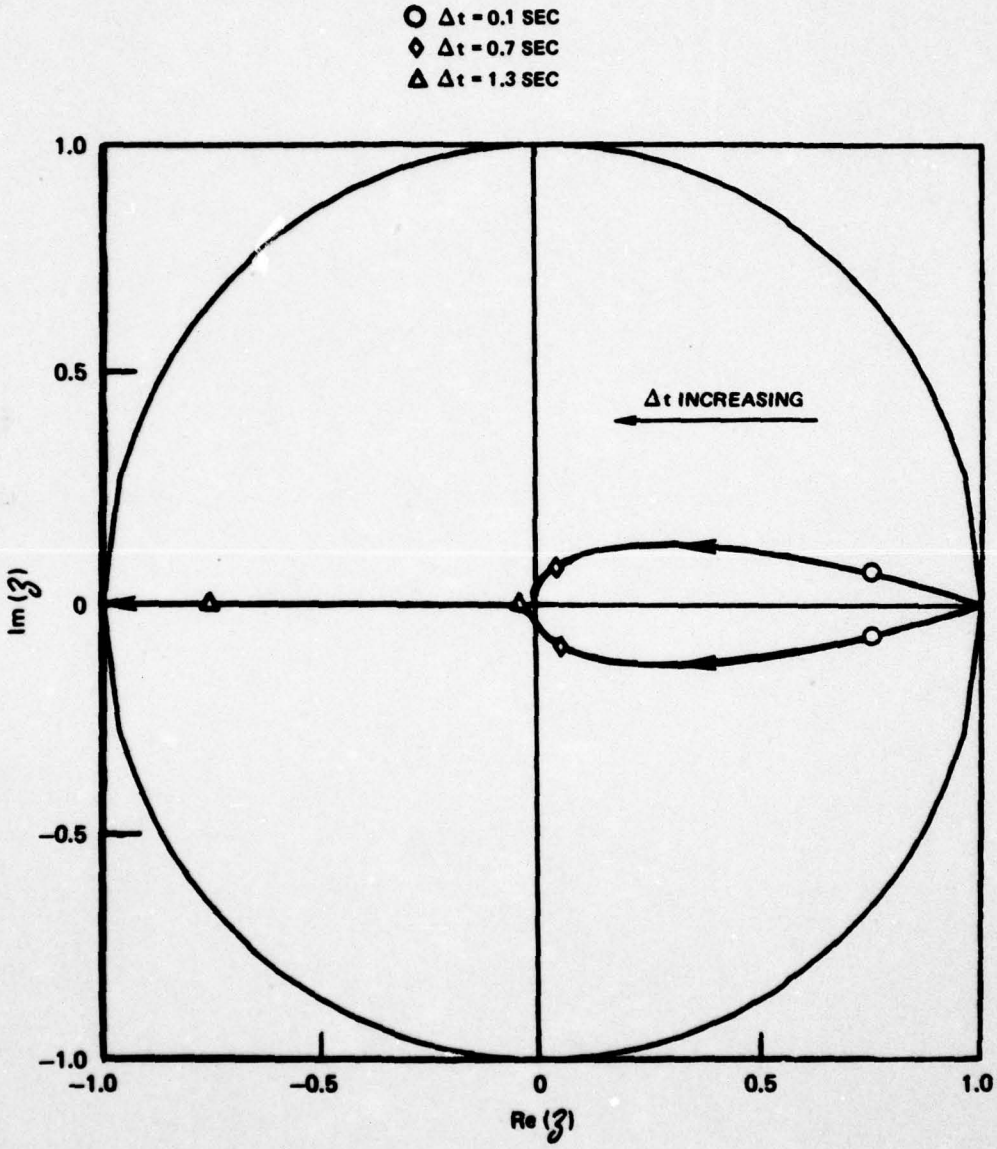
- 36 BITS, JORDAN CANONICAL STRUCTURE
- 6 BITS, JORDAN CANONICAL STRUCTURE
- 6 BITS, COMPANION STRUCTURE



79–03–46–5

SECOND–ORDER SYSTEM z–PLANE POLES AS A FUNCTION OF SAMPLE TIME

DIGITAL CONTROLLER WITH STANDARD STRUCTURE AND 8 BIT WORD LENGTH



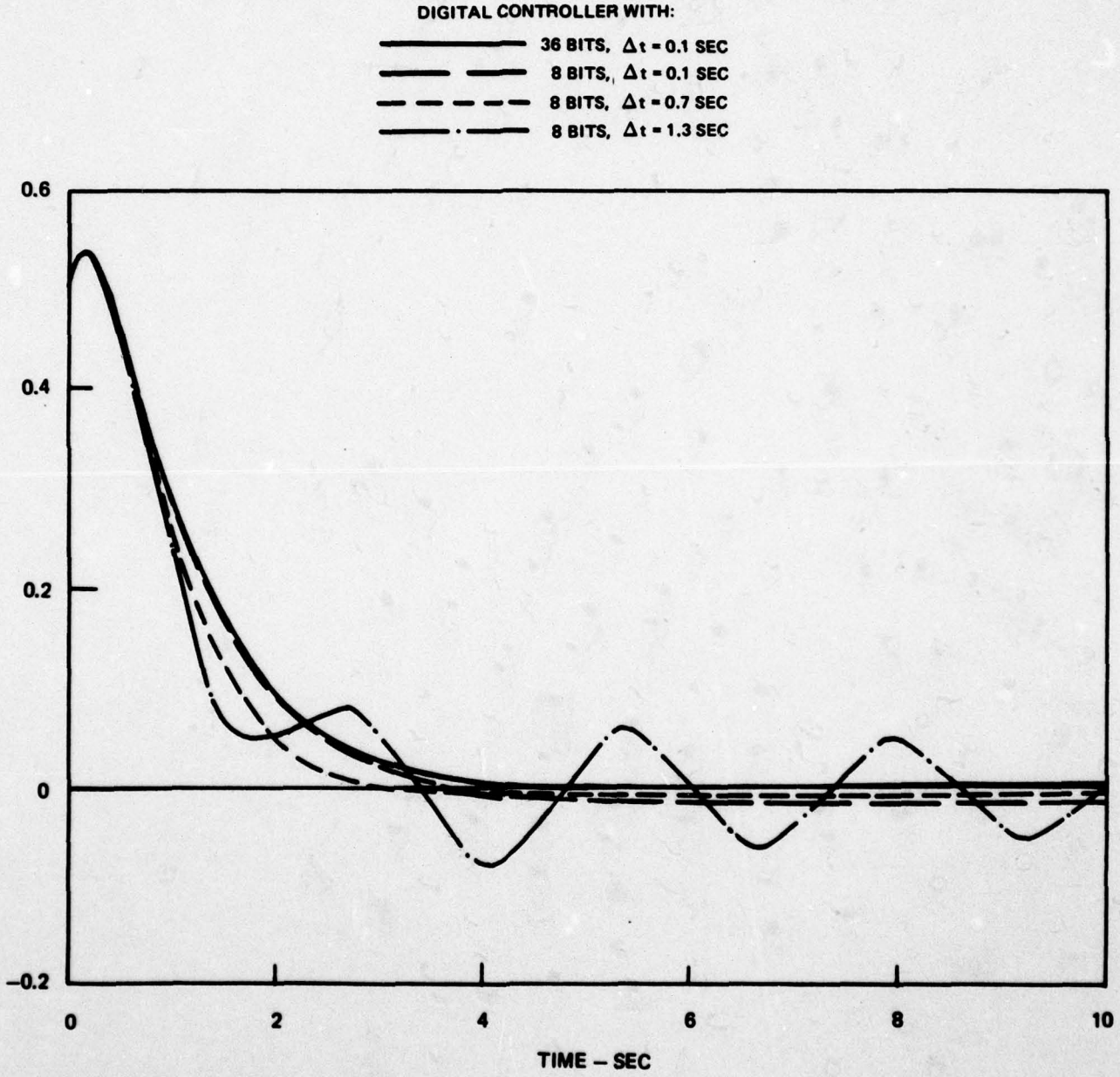
79–03–86–5

FIG. 13

SECOND–ORDER OUTPUT RESPONSE AS A FUNCTION OF SAMPLE TIME

RESPONSE TO INITIAL CONDITION: $X_0' = (0.5, 0.5)$

STANDARD STRUCTURE WITHIN CONTROLLER

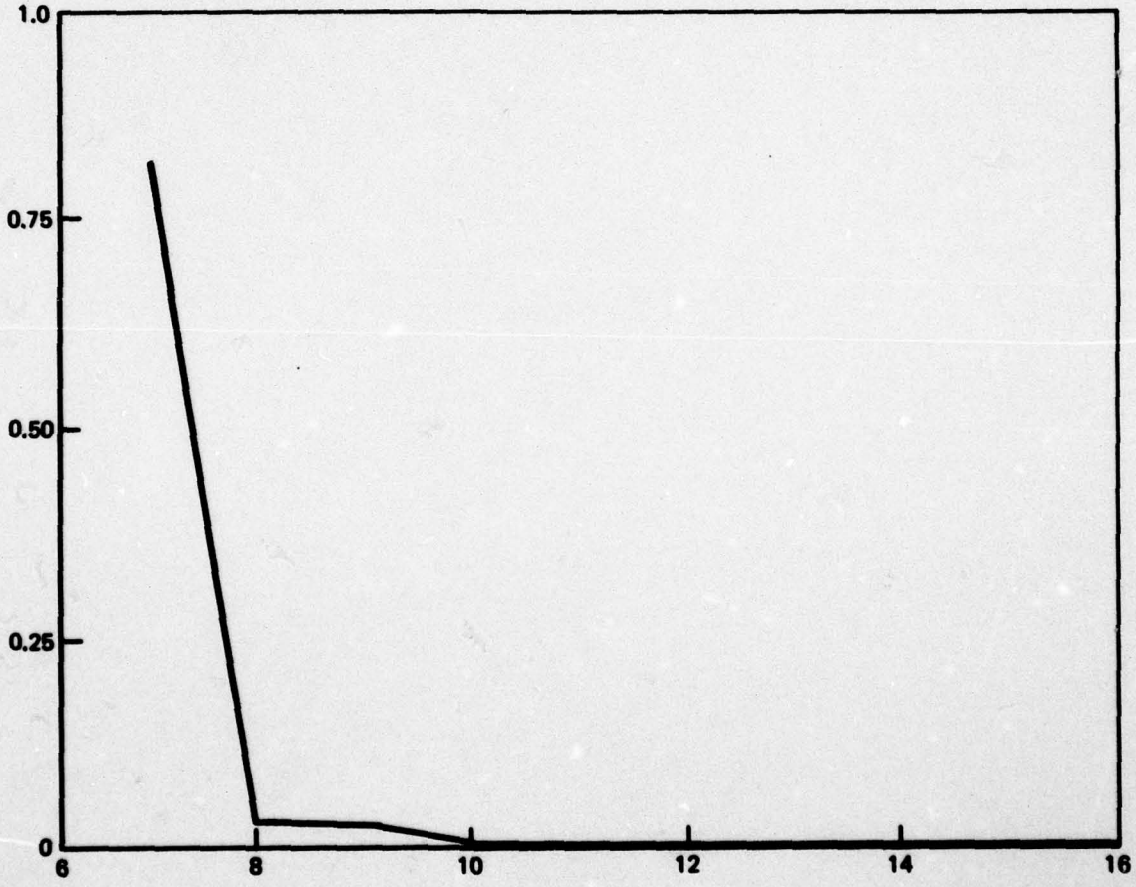


79–03–46–6

FIG. 14

NORMALIZED PERFORMANCE INDEX FOR LINEARIZED F100 ENGINE MODEL

STANDARD STRUCTURE WITHIN CONTROLLER



79–03–198–1

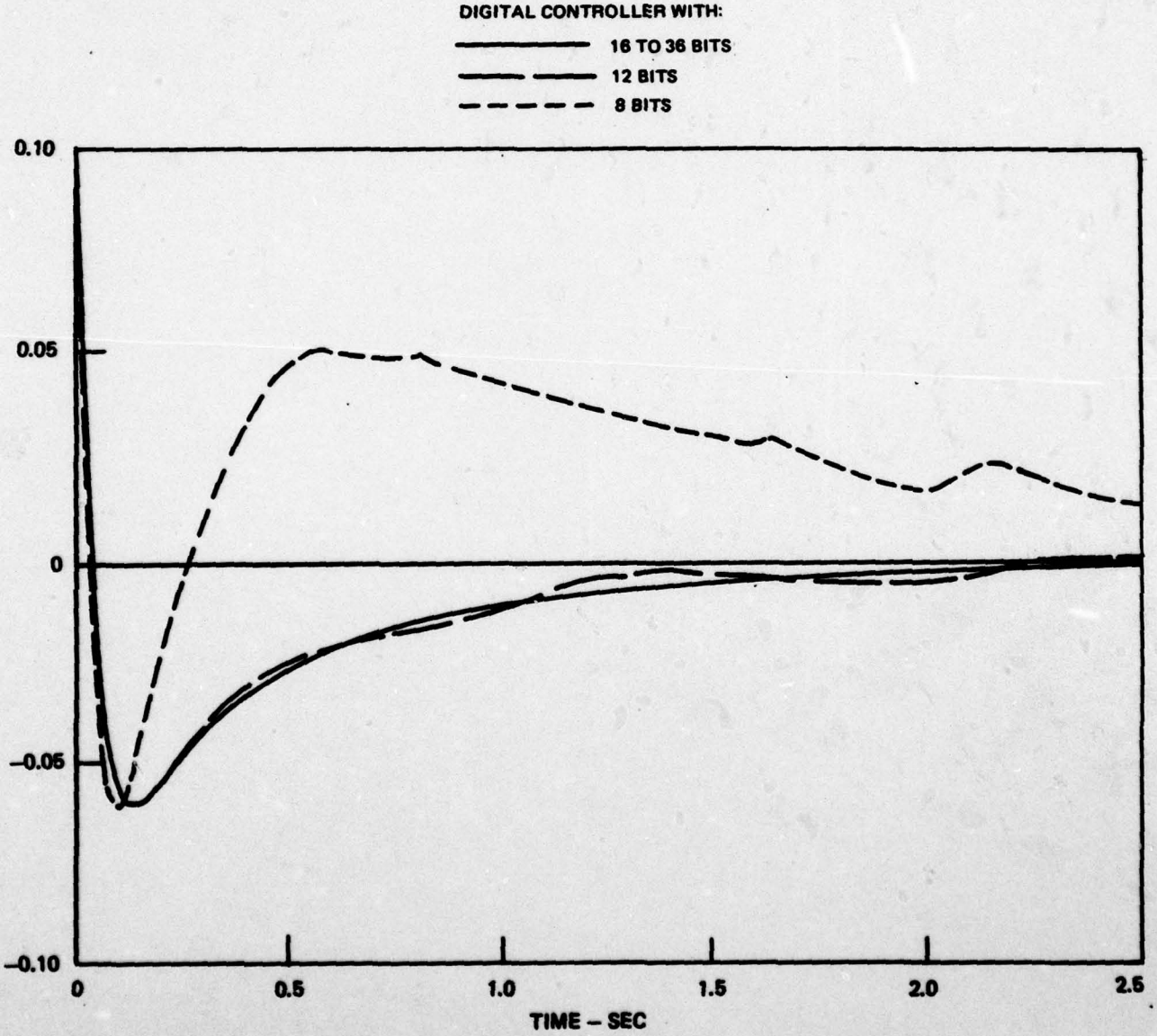
FIG. 15

F100 ENGINE MODEL PERTURBATIONAL AFTERBURNER PRESSURE RESPONSE AS A FUNCTION OF WORD LENGTH

RESPONSE TO INITIAL CONDITION: $X_0' = (0.1, 0.1, 0.1, 0.1, 0.1)$

STANDARD STRUCTURE WITHIN CONTROLLER

$\Delta t = 0.025$ SEC

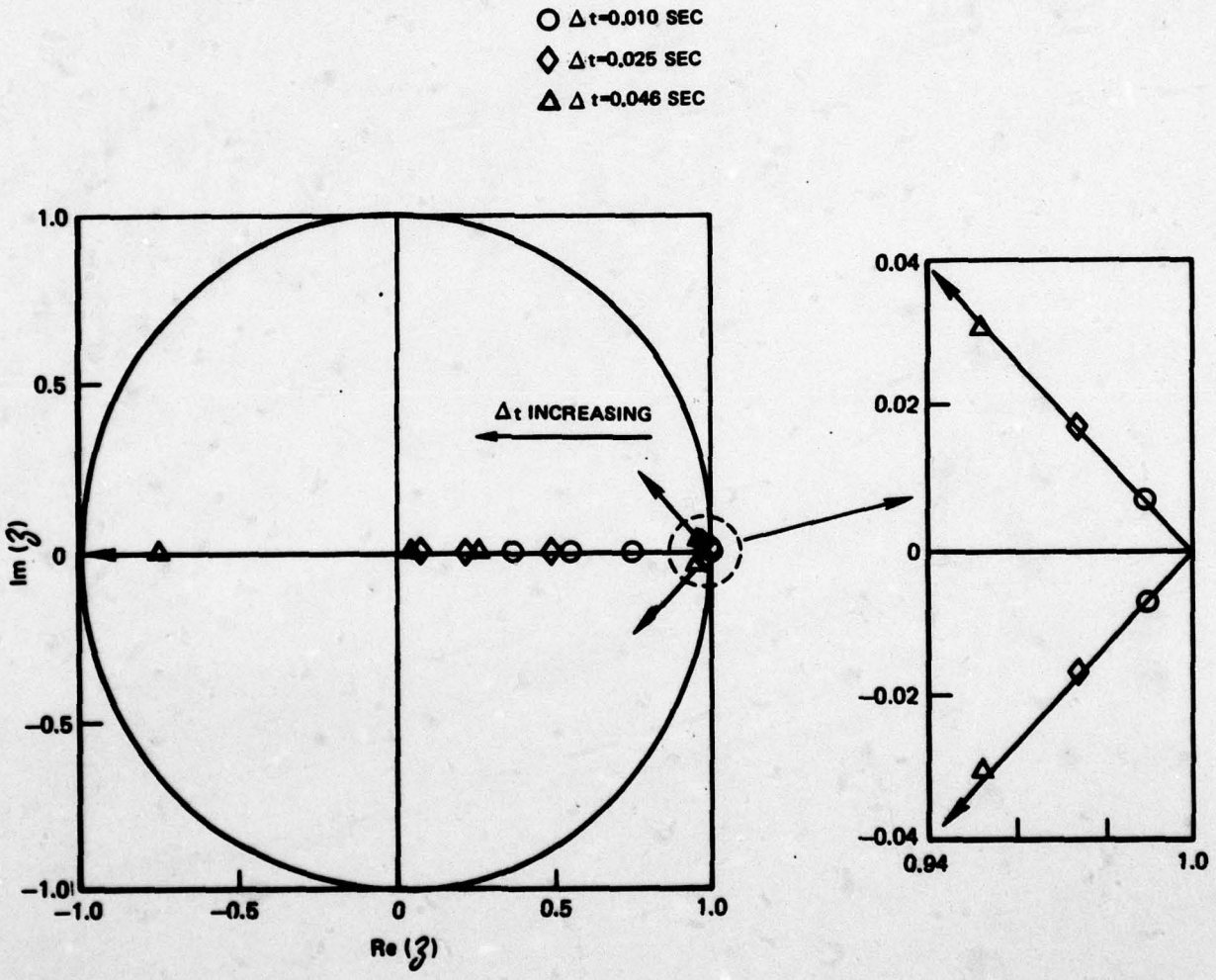


79–03–46–3

FIG. 16

F100 ENGINE MODEL $\mathcal{Z}$–PLANE POLES AS A FUNCTION OF SAMPLE TIME

DIGITAL CONTROLLER WITH STANDARD STRUCTURE AND 12 BIT WORD LENGTH



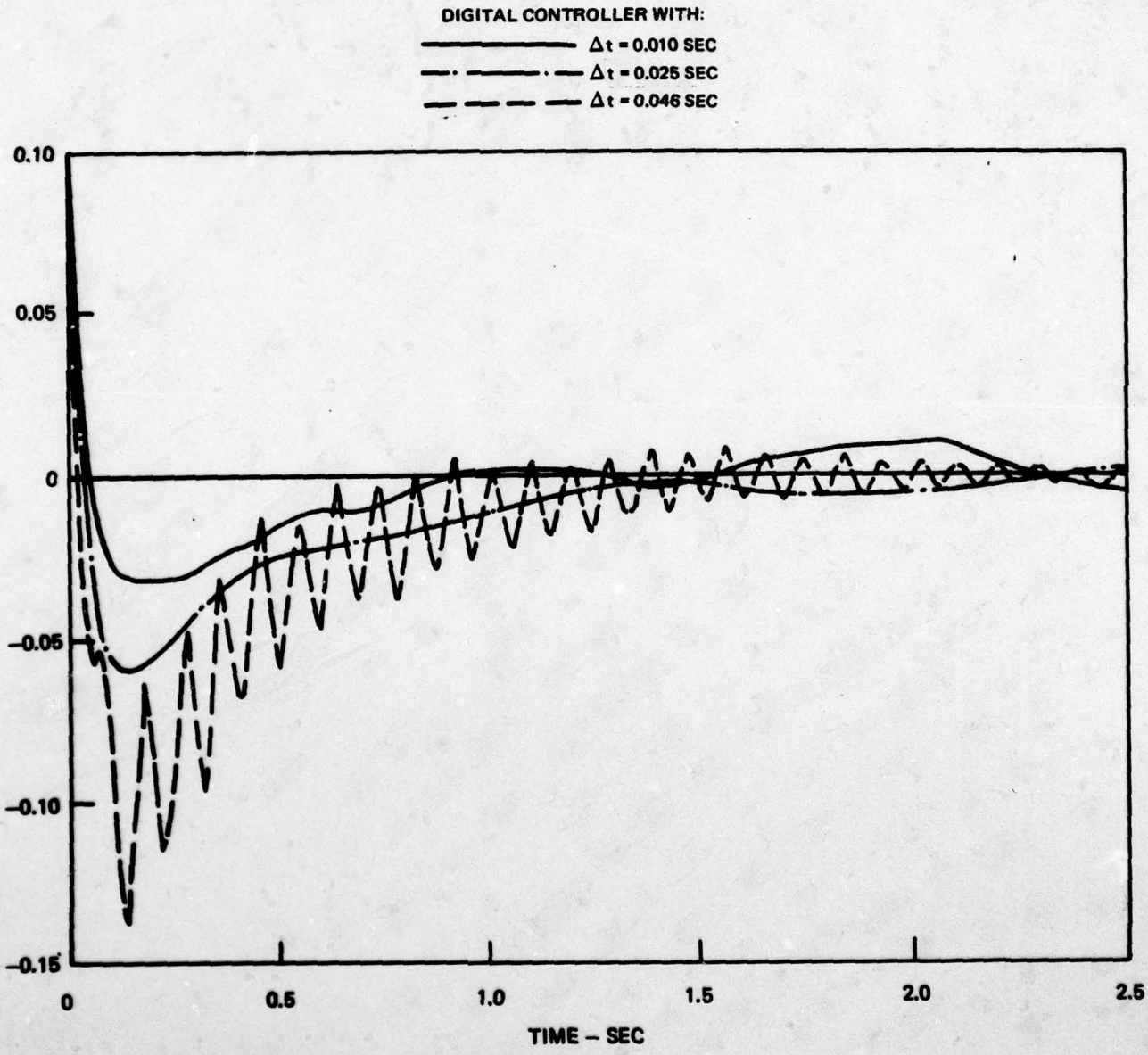
79–03–46–2

FIG. 17

F100 ENGINE MODEL PERTURBATIONAL AFTERBURNER PRESSURE RESPONSE AS A FUNCTION OF SAMPLE TIME

RESPONSE TO INITIAL CONDITION: $X'_0 = (0.1, 0.1, 0.1, 0.1, 0.1)$

DIGITAL CONTROLLER WITH STANDARD STRUCTURE AND 12 BIT WORD LENGTH



79-03-46-4

APPENDIX

MICROPROCESSOR SURVEY AND COMPUTER CODE

Characteristics of (1) microprocessors, (2) A/D and D/A converters, and (3) hardware multipliers are presented in this Appendix. The characteristics tabulated here were obtained from Electrical Design News (EDN) 1976-1978 as well as from TRW product sheets. Microprocessor characteristics -- including word length, internal registers, indexed addressing capabilities, and multiply instruction capability -- are listed in Table A-I. The A/D and D/A characteristics -- including word length, conversion time, and technology -- are shown in Table A-II. Table A-III displays multiplier characteristics including word length, multiply time, and technology.

In addition, microprocessor code required for calculating the matrix/vector multiplication computation times and code memory requirements are presented in this Appendix. Codes for implementing linear quadratic Gaussian control logic on an Intel 8080 microprocessor and on a DEC LSI 11/2 microprocessor are shown in Figs. A-1 and A-2, respectively. The number of instructions for the Intel 8080 microprocessor is more than double the number of instructions for the DEC LSI 11/2 due primarily to the limited addressing capability of the Intel 8080 microprocessor.

TABLE A-I

REPRESENTATIVE MICROPROCESSORS

R79-9442S8-2

Characteristics / Microprocessor	Second Source	Technology	Word Length--Data; Address (Bits)	On-Chip Clock	Internal Register	Power Supplies (Volts)	Indexed Addressing	Instruction Execution Time (μsec)	Multiply Instruction	ADC On-Chip
Intel 8080/8085	Y	NMOS	8;16	N/Y	7	±5,+12/+5	N	2.0-8.5	N	N
Zilog Z80	Y	NMOS	8;16	N	12	+5	Y	2.0	N	N
Motorola 6800/6802	Y	NMOS	8;16	N/Y	2	+5	Y	2.0-5.0	N	N
Zilog Z8	N	NMOS	8;16	Y	124	+5	Y	1.5-2.2	N	N
Mostek 3870	Y	NMOS	8;16	Y	64	+5	Y	2.0	N	N
Intel 8048 (Family)	Y	NMOS	8;16	Y	-	+5	-	1.4-10.0	N	Y(1)
TI 9900/SBP 9900A	Y	NMOS/I^2L	8,16;16/8;14	Y	16	+5	Y	5.0-10.0	Y	N
Intersil IM6100	Y	CMOS	12;12	N	-	+5	N	5.0	N	N
National Semiconductor 8900	N	NMOS	16;16	N	4	±5,+12	Y	10.0	N	N
Intel 8086	N	NMOS	16;16	N	7	+5	Y	0.6-3.4	Y	N
Zilog Z8000	Y	NMOS	16;24	N	16	+5	Y	0.7-17.5	Y	N
Motorola 68000	-	NMOS	16;24	N	16	+5	Y	-	Y	N
Intel 3000	Y	Bipolar	2 BIT SLICE	N	-	+5	-	-	-	N
Motorola 10800	-	Bipolar/ECL	4 BIT SLICE	N	-	-5.2,-2.0	-	-	-	N
DEC LSI 11/2	-	-	16;16	N	8	-	Y	2.0-232.0	Y	N

(1) Intel 8022 only

TABLE A-II

REPRESENTATIVE A/D AND D/A CONVERTERS

Converter Type	Manufacturer	Model	Word Length (bits)	Conversion Time (μsec)	Technology
A/D	TRW	TDC1007J	8	35×10^{-3}	Bipolar
	TRW	TDC1001J	8	400×10^{-3}	Bipolar
	TRW	TDC1002J	8	1	Bipolar
	Analog Devices	AD75705	8	40	CMOS
	Datel	ADC-MC88C	8	500	Bipolar
	Analog Devices	AD7570L	10	120	CMOS
	Datel	ADC-HX12B	12	20	Hybrid
	Analog Devices	AD572BD	12	25	---
	Micre Networks	ADC80	12	25	Hybrid
	National Semiconductor	ADC1210	12	50	Hybrid
D/A	TRW	TDC1016J	8	35×10^{-3}	Bipolar
	Analog Devices	AD7523JN	8	100×10^{-3}	---
	Datel	DAC-UP88	8	2	Bipolar
	National Semiconductor	DAC0800	8	135	Bipolar
	Datel	DAC-088	8	150	Bipolar
	TRW	TDC1017J	10	50×10^{-3}	Bipolar
	Analog Devices	AD7541KN	12	1	---
	Datel	DAC-HK12B	12	3	Hybrid
	Harris Semiconductor	H1-5612	12	85	Bipolar
	Harris Semiconductor	H1-562	12	200	Bipolar
	Datel	DAC-HA12B	12	500	Hybrid
	Analog Devices	AD7531	12	500	Hybrid

TABLE A-III

REPRESENTATIVE MULTIPLIERS

Manufacturer	Model	Word Length (bits)	Multiply Time (nsec)	Technology	Accumulator
TRW	TDC1008J	8	70	TTL	Y
MONOLITHIC MEMORIES	57558	8	100	—	N
TRW	MPY-8AJ	8	130	TTL	N
TRW	MPY-12A	12	150	TTL	N
TRW	TDC1003J	12	175	TTL	Y
TRW	TDC1010J	16	115	TTL	Y
TRW	MPY-16A	16	160	TTL	N
AMD	9511	16	42000	—	N

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

PRELIMINARY INTEL 8080 SOFTWARE FOR LQG CONTROLLER

```
1:          ;KALMAN FILTER/CONTROLLER SOFTWARE--PRELIMINARY
2:          ;DEVELOPED BY R. S. EIDENS
3:          ;UTRC---1978
4:;
5:;
6:          ORG 4000H                    ;STARTING ADDRESS FILTER--VERSION 1.0
7:          N1 EQU 5011H                 ;POINTER TO  HZ PARTIAL RESULTS
8:          N2 EQU 502CH                 ;POINTER TO  PHIX PARTIAL RESULTS
9:          N3 EQU 5041H                 ;POINTER TO  GX PARTIAL RESULTS
10:         NSS EQU 5012H                ;POINTER TO HZ
11:         NSS1 EQU 502DH               ;POINTER TO PHIX
12:         NV EQU 5030H                 ;POINTER TO VECTORS
13:         ST2 EQU 5014H                ;START ADDR. FOR PHIX CONSTANTS
14:         ST3 EQU 5035H
15:         ASTATE EQU 5036H             ;POINTER TO X(K+1/K+1)
16:         MU EQU 5045H
17:         NS EQU 3                     ;NO. OF STATES
18:         NM EQU 2                     ;NO. OF MEASUREMENTS
19:         NC EQU 1                     ;NO. OF CONTROLS
20:         NC1 EQU 2
21:         N2M2 EQU 8
22:         N2M3 EQU 0
23:         N2M1 EQU 5                   ;DISPLACEMENT
24:         ST1 EQU 5000H                ;STARTING ADDR.OF CONSTANTS
25:         REL EQU 50H
26:         MVI A,NM                     ;SIZE OF MEASUREMENT VECTOR INTO ACC.
27:         LXI H,ST1                    ;GET STARTING ADDRESS OF CONSTANTS
28:NRLP:    INX H                        ;NEXT ARGUMENT ADDR.
29:         MVI B,NS                     ;SIZE OF STATE VECTOR
30:         MOV C,M                      ;GET MULTIPLIER
31:NCLP:    INX H                        ;NEXT ADDR.
32:         MOV E,M                      ;GET MULTIPLICAND
33:         PUSH B                       ;STORE NM,ADDR. POINTER,FLAGS AND NS
34:         PUSH H
35:         PUSH PSW
36:         CALL MULT                    ;(C)*(E)=>(DE)
37:         MOV A,E                      ;RESULT INTO ACC.(TRUNCATE)
38:         LXI H,N1                     ;ADDR.OF STORAGE
39:         MOV E,M                      ;GET ADDRESS
40:         MVI D,REL                    ;EXTEND DE REG FOR ADDR.
41:         STAX D                       ;STORE RESULT
42:         MOV A,E
43:         ADI NM                       ;ADD DISPLACEMENT
44:         MOV M,A                      ;NEW ADDR.
45:         POP PSW                      ;RESTORE FLAGS,NM,CONSTANTS LOCATION,NS
46:         POP H
47:         POP B
48:         DCR B                        ;TEST LOOP
49:         JNZ NCLP
50:         PUSH PSW                     ;SAVE FLAGS,ACC.
51:         PUSH H
52:         LXI H,N1                     ;SET UP FOR NEXT MEASUREMENT
53:         MOV A,M

54:         SUI N2M1
55:         MOV M,A
56:         POP H
57:         POP PSW                      ;TEST OUTER LOOP
58:         DCR A
59:         JNZ NRLP
60:         MVI A,NS                     ;PREPARE FOR ADDITION
61:         INX H
62:         PUSH B                       ;COMPLETE MATRIX-VECTOR MULTIPLY BY ADDING
63:SMLP:    PUSH PSW
64:         MVI C,NM                     ;ADDITION SUBROUTINE
65:         CALL SUM
66:         PUSH H                       ;PREPARE FOR STORING RESULT
67:         LXI H,NSS
68:         MOV E,M                      ;EXTEND FOR ADDR.
69:         MVI D,REL                    ;STORE FINAL RESULT
70:         STAX D                       ;NEW ADDR.
71:         INX D
72:         INX D
73:         MOV M,E                      ;STORE NEW ADDR.
74:         POP H                        ;RESTORE REGISTERS
75:         POP PSW
```

(CONTINUED)

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

PRELIMINARY INTEL 8080 SOFTWARE FOR LQG CONTROLLER

(CONTINUED)

```
 76:             DCR A
 77:             JNZ SMLP                ;SUM LOOP
 78:             POP B                   ;HZ PRODUCT COMPLETE
 79:             MVI A,NS                ;PREPARE FOR PHIX(K/K) PRODUCT
 80:             LXI H,ST2               ;POINTER TO PHI
 81:NRLP1:       INX H
 82:             MVI B,NS                ;SIZE OF STATE VECTOR
 83:             MOV C,M                 ;GET MULTIPLIER
 84:NCLP1:       INX H
 85:             MOV E,M                 ;GET MULTIPLICAND
 86:             PUSH B
 87:             PUSH H
 88:             PUSH PSW
 89:             CALL MULT               ;(C) * (E) => (DE)
 90:             MOV A,E                 ;PREPARE TO STORE RESULT
 91:             LXI H,N2
 92:             MOV E,M
 93:             MVI D,REL
 94:             STAX D                  ;STORE RESULT
 95:             MOV A,E                 ;REPEAT OF LINES 42-47
 96:             ADI NS
 97:             MOV M,A
 98:             POP PSW
 99:             POP H
100:             POP B
101:             DCR B
102:             JNZ NCLP1
103:             PUSH PSW
104:             PUSH H
105:             LXI H,N2
106:             MOV A,M
107:             SUI N2M2
108:             MOV M,A
109:             POP H
110:             POP PSW
111:             DCR A
112:             JNZ NRLP1
113:             MVI A,NS

114:             INX H
115:             PUSH B
116:SMLP1:       PUSH PSW
117:             MVI C,NS
118:             CALL SUM
119:             PUSH H
120:             LXI H,NSS)
121:             MOV E,M
122:             MVI D,REL
123:             STAX D
124:             INX D
125:             INX D
126:             MOV M,E
127:             POP H
128:             POP PSW
129:             DCR A
130:             JNZ SMLP1
131:             POP B                   ;PHIX(K/K) COMPLETE
132:             LXI H,NY                ;PREPARE FOR VECTOR ADD
133:             MVI B,NS
134:             LXI D,ASTATE            ;POINTER TO PREDICT SOLUTION
135:             CALL VSUM
136:             MVI A,NS                ;PREPARE FOR GX(K+1/K+1) MPY
137:             LXI H,ST3
138:NRLP3:       INX H                   ;NEXT ARGUMENT ADDR.
139:             MVI B,NC                ;COLUMN COUNTER
140:             MOV C,M                 ;GET MULTIPLIER
141:NCLP3:       INX H                   ;NEXT ADDR.
142:             MOV E,M                 ;GET MULTIPLICAND
143:             PUSH B                  ;STORE NS,ADDR. POINTER,FLAGS AND NC
144:             PUSH H                  ;
145:             PUSH PSW                ;
146:             CALL MULT               ;(C)*(E)=>(DE)
147:             MOV A,E                 ;RESULT INTO ACC.(TRUNCATE)
148:             LXI H,N3                ;ADDR. OF STORAGE
149:             MOV E,M                 ;GET ADDRESS
150:             MVI D,REL               ;EXTEND DE REG FOR ADDR.
```

(CONTINUED)

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

PRELIMINARY INTEL 8080 SOFTWARE FOR LQG CONTROLLER (CONTINUED)

```
151:            STAX D              ;STORE RESULT
152:            MOV A,E             ;
153:            ADI NC              ;ADD DISPLACEMENT
154:            MOV M,A             ;NEW ADDR. IN LOCATION 17
155:            POP PSW             ;RESTORE FLAGS,NC,CONSTANTS LOCATION,NR
156:            POP H               ;
157:            POP B               ;
158:            DCR B               ;TEST LOOP
159:            JNZ NCLP3           ;
160:            PUSH PSW            ;
161:            PUSH H              ;
162:            LXI H,N3            ;
163:            MOV A,M             ;
164:            SUI N2M3            ;
165:            MOV M,A             ;
166:            POP H               ;
167:            POP PSW             ;
168:            DCR A               ;TEST OUTER LOOP
169:            JNZ NRLP3           ;
170:            MVI A,NC            ;PREPARE FOR ADDITION
171:            INX H               ;
172:            PUSH B              ;
173:SMLP4:      PUSH PSW            ;
174:            MVI C,NS            ;
175:            CALL SUM            ;
176:            PUSH H              ;
177:            LXI H,NU            ;
178:            MOV E,M             ;
179:            MVI D,REL           ;EXTEND FOR ADDR.
180:            STAX D              ;STORE FINAL RESULT
181:            INX D               ;NEW ADDR.
182:            MOV M,E             ;STORE NEW ADDR.
183:            POP H               ;
184:            POP PSW             ;
185:            DCR A               ;
186:            JNZ SMLP4           ;SUM LOOP
187:            POP B               ;COMPLETE GX(K+1/K+1) = U(K+1/K+1)
188:            JMP DONE            ;
189:                                ;
190:                                ;
191:                                ;
192:                                ;
193:MULT:       MOV A,C             ;CHECK SIGN OF (C)
194:            ORA A               ;
195:            JM MUL5             ;(-C)*(-E)
196:            MOV A,E             ;CHECK SIGN OF (E)
197:            ORA A               ;
198:            JM MUL3             ;(E)=>-(E)
199:MUL1:       CALL IMUL           ;MAG(C)*MAG(E)=>(DE)
200:MUL2:       ORA A               ;
201:            RET                 ;
202:MUL3:       DCR E               ;2'S COMP. (E)
203:            MOV A,E             ;
204:            CMA                 ;
205:            MOV E,A             ;
206:MUL4:       CALL IMUL           ;(-C)*(E)
207:C2D2:       DCX D               ;2'S COMP. (DE)
208:            MOV A,E             ;1'S COMP. (DE)
209:            CMA                 ;
210:            MOV E,A             ;
211:            MOV A,D             ;
212:            CMA                 ;
213:            MOV D,A             ;
214:            JMP MUL2            ;
215:MUL5:       DCR C               ;2'S COMPLEMENT OF (C)
216:            MOV A,C             ;
217:            CMA                 ;
218:            MOV C,A             ;
219:            MOV A,E             ;GET MULTIPLIER
220:            ORA A               ;CHECK SIGN OF (E)
221:            JP MUL4             ;(-C)*(E)
222:            DCR E               ;2'S COMP. (E)
223:            MOV A,E             ;
224:            CMA                 ;
225:            MOV E,A             ;
```

(CONTINUED)

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

PRELIMINARY INTEL 8080 SOFTWARE FOR LOG CONTROLLER (CONTINUED)

```
226:          JMP   MUL1
227: IMUL:    MVI   D,0H        ;EXTEND E REGISTER
228:          MOV   A,C         ;GET MULTIPLIER
229:          LXI   H,0H        ;CLEAR HL
230:          MVI   B,8         ;COUNTER
231: IMUL1:   DAD   H           ;ROTATE HL LEFT
232:          RAL               ;ACC LEFT THRU CARRY
233:          JNC   CHNT        ;CARRY=?1

234:          DAD   D           ;YES,PROD=PROD+MULTIPLICAND
235: CHNT:    DCR   B           ;
236:          JNZ   IMUL1       ;
237:          XCHG              ;PUT RESULT IN DE
238:          RET               ;
239: SUM:     XRA   A           ;CLEAR ACC.
240: SUMD:    ADD   M           ;ADD(HL) TO ACC
241:          INX   H           ;
242:          DCR   C           ;
243:          JNZ   SUMD        ;
244:          RET               ;RETURN--NEXT TIME INCREMENT
245: VSUM:    XRA   A
246: VSUMD:   ADD   M
247:          INX   H
248:          ADD   M
249:          INX   H
250:          STAX  D
251:          MOV   A,E
252:          ADI   NC1
253:          MOV   E,A
254:          DCR   B
255:          JNZ   VSUM
256:          RET
257: DONE:    NOP
258:          END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

PRELIMINARY DEC LSI 11/2 SOFTWARE FOR LQG CONTROLLER

```
1:          ;DEC LSI 11/2 CODE FOR
2:          ;KALMAN FILTER/CONTROLLER--PRELIMINARY
3:          ;DEVELOPED BY R. S. EIDENS
4:          ;UTRC---1979
5::
6::
7:          MOV  POINTER,R0          ;POINTER TO SIZE OF H-ARRAY
8:          MOV  MM,(R0)             ;SIZE OF H-ARRAY
9:M:        MOV  SM,R3               ;START ADDR. OF H-ARRAY (KALMAN GAINS)
10:N:       MOV  M,2(R0)             ;SIZE OF Z-ARRAY
11:         MOV  S2,R1               ;START ADDR. OF Z-ARRAY
12:         MOV  (R0),-(R6)          ;PUSH R0
13:MMUL1:   JSR  PC,MULT             ;SCALOR MPY.,RESULT IN R2
14:         MOV  (R6)+,(R0)          ;POP R0
15:         MOV  PROD,1000(R3)       ;ADDR. OF MZ PRODUCTS
16:         DEC  @R0                 ;DONE?
17:         BEQ  CT1                 ;YES,CONTINUE
18:         DEC  2(R0)               ;NO,UPDATE COUNTER
19:         BNE  MMUL1               ;DONE ROW?
20:         BR   Z                   ;YES,NEXT ROW
21:CT1:     MOV  SP1,R1              ;START ADDR. OF PRODUCTS
22:         MOV  SR1,R2              ;START ADDR. OF RESULTS
23:         MOV  N,R0                ;SIZE OF X-VECTOR
24:LOOP1:   MOV  M,R3                ;SIZE OF Z-VECTOR
25:         CLR  R4
26:LOOP2:   ADD  (R1)+,R4            ;SCALOR ADD, RESULTS IN R4
27:         DEC  R3
28:         BNE  LOOP2               ;DONE?
29:         MOV  R4,(R3)+            ;STORE RESULTS
30:         DEC  R0
31:         BNE  LOOP1               ;DONE WITH MZ PRODUCT
32:         MOV  POINTER,R0          ;POINTER TO SIZE OF D-ARRAY
33:         MOV  MM,(R0)             ;SIZE OF PHI-ARRAY
34:PHI:     MOV  SPHI,R3             ;START ADDR. OF PHI (SYSTEM MATRIX)
35:XHAT:    MOV  N,2(R0)             ;SIZE OF X-VECTOR
36:         MOV  S1,R1               ;START ADDR. OF XHAT
37:         MOV  (R0),-(R6)          ;PUSH R0
38:MMUL2:   JSR  PC,MULT             ;SCALOR MPY.,RESULT IN R2
39:         MOV  (R6)+,(R0)          ;POP R0
40:         MOV  PROD,1000(R3)       ;STORE RESULTS
41:         DEC  @R0                 ;DONE?
42:         BEQ  CT2                 ;YES
43:         DEC  2(R0)
44:         BNE  MMUL2               ;NO
45:CT2:     MOV  SP2,R1              ;START ADDR. OF PRODUCTS
46:         MOV  SR2,R2              ;START ADDR. OF RESULTS
47:         MOV  N,R0                ;SIZE OF X-VECTOR
48:LOOP3:   MOV  N,R3                ;SIZE OF X-VECTOR
49:         CLR  R4
50:LOOP4:   ADD  (R1)+,R4            ;SCALOR ADD, RESULT IN R4
51:         DEC  R3
52:         BNE  LOOP4               ;DONE?
53:         MOV  R4,(R2)+            ;STORE RESULT
54:         DEC  R0
55:         BNE  LOOP3               ;DONE WITH PHI XHAT
56:         MOV  SR1,R1              ;PREPARE FOR VECTOR ADD
57:         MOV  SP2,R2
58:         MOV  N,R3
59:SUM:     ADD  (R1)+,(R2)+         ;VECTOR ADD
60:         DEC  R3
61:         BNE  SUM                 ;DONE WITH VECTOR ADD?
62:         MOV  POINTER,R0          ;REPEAT LINES 7 TO 31
63:         MOV  NL,(R0)             ;CONTROL DIMENSION
64:G:       MOV  SG,R3               ;START ADDR. OF G (CONTROL GAINS)
65:XN:      MOV  N,2(R0)
66:         MOV  SP2,R1
67:         MOV  (R0),-(R6)
68:MMUL3:   JSR  PC,MULT
69:         MOV  (R6)+,(R0)
70:         MOV  PROD,1000(R3)
71:         DEC  @R0
72:         BEQ  CT3
73:         DEC  2(R0)
74:         BNE  MMUL3
75:         BR   XN
```

(CONTINUED)

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

PRELIMINARY DEC LSI 11/2 SOFTWARE FOR LOG CONTROLLER (CONTINUED)

```
 76:CT?:     MOV  SP3,R1
 77:         MOV  SP3,R2
 78:         MOV  NC,R0
 79:LOOP5:   MOV  N,R3
 80:         CLR  R4
 81:LOOP6:   ADD  (R1)+,R4
 82:         DEC  R3
 83:         BNE  LOOP6
 84:         MOV  R4,(R2)+
 85:         DEC  R0
 86:         BNE  LOOP5                ;U(K+1/K+1) COMPLETED?
 87:MULT:    MOV  (R1)+,R2             ;CHECK SIGN MULTIPLIER
 88:         BMI  MUL5
 89:         MOV  (R3)+,R4             ;CHECK SIGN MULTIPLICAND
 90:         BMI  MUL3
 91:MUL1:    JSR  PC,IMUL              ;INTEGER, UNSIGNED MPY.,RESULT POSITIVE
 92:MUL2     RST  PC
 93:MUL3:    DEC  R4                   ;2'S COMPLIMENT MULTIPLICAND
 94:         COM  R4
 95:MUL4:    JSR  PC,IMUL              ;INTEGER, UNSIGNED MPY.,RESULT WRONG SIGN
 96:         DEC  R2                   ;2'S COMPLIMENT RESULT
 97:         COM  R2
 98:         BR  MUL2
 99:MUL5:    DEC  R2                   ;2'S COMPLIMENT MULTIPLIER
100:         COM  R2
101:         MOV  (R3)+,R4             ;CHECK SIGN OF MULTIPLICAND
102:         BLP  MUL4
103:         DEC  R4                   ;2'S COMPLIMENT MULTIPLICAND
104:         COM  R4
105:         BR  MUL1                  ;END OF MULTIPLY MAIN
106:IMUL:    MOV  #0,PROD              ;PREPARE 32 BIT REGISTER FOR PRODUCT
107:         MOV  #0,PROD1
108:         MOV  #16.,R0              ;COUNTER
109:IMUL1:   ASL  PROD1                ;SHIFT 32 BITS LEFT
110:         ASL  PROD
111:         ADC  PROD1                ;ADJUST CARRY
112:         BOL  R2                   ;CHECK FOR 1'S IN MULTIPLIER
113:         BCC  CHNT                 ;1 IN MULTIPLIER?


114:         ADD  (R4),PROD            ;YES,PRODUCT=PRODUCT + MULTIPLIER
115:         ADC  PROD1                ;ADJUST CARRY
116:CHNT:    DEC  R0                   ;DONE?
117:         BNE  IMUL1                ;NO
118:         RTS  PC                   ;YES
119:         END
```

Distribution List
Report R79-944258-2
Contract F49620-78-C-0017

Director of Mathematical and Information Sciences
Bldg. 410
Bolling AFB, Washington, D. C. 20332
Attn: Major Charles L. Nefzger (16)

ASD/XRT
Wright-Patterson AFB, OH 45433
Attn: Lt Col W. Othling

OASD/PA&E
Pentagon
Washington, DC 20360
Attn: Mr. T. P. Christie

NASA-Flight Research Center
Edwards AFB, CA 93523
Attn: Mr. H. A. Rediess

SAMSO/YAD
Technology-Guidance & Control
Los Angeles AFS, CA 90045

NASA-Ames Research Center
Moffett Field, CA 94035
Attn: Mr. Elwood C. Stewart

Naval Ocean Systems Center
Tactical Command Control Division
San Diego, CA 92152
Attn: Dr. Robert Kolb, Code 824

NASA-Langely Research Ctr
Hampton, VA 23365
Attn: Mr. Larry W. Taylor, Jr.

AFIT/ENE
Wright-Patterson AFB, OH 45433
Attn: Capt. Gary Reid

Office of Naval Research
Technology Group
800 North Quincy St.
Arlington, VA 22217
Attn: Mr. David Siegel

AFIT/ENE
Wright-Patterson AFB, OH 45433
Attn: Capt James Negro

AFFDL/FGL
Wright-Patterson AFB, OH 45433
Attn: Paul Blatt

RADC/ISCP
Griffiss AFB, NY 13441
Attn: Heywood Webb

ARO
P.O. Box 12,211
Research Triangle Park, NC 27709
Attn: Dr. J. Chandra

Office of Naval Research
Mathematics Group
800 North Quincy Street
Arlington, VA 22217
Attn: Dr. Stewart Brodsky

AFAPL/TBC
Wright-Patterson AFB, OH 45433
Attn: Mr. Charles Skira

AFATL/DLY
Eglin AFB, FL 32542
Attn: Dr. Jesse Gonzales

AFWL/LRO
Kirtland AFB, NM 87117
Attn: Lt Col Dale Neal

David Taylor Naval Ship R&D Center
Code 2730
Annapolis, MD 21404
Attn: Mr. Walter J. Blumberg

AFWL/AL
Kirtland AFB, NM 87117
Attn: Maj Kenneth Herring