Digital Control Systems Implementation and Computational Techniques

ACADEMIC PRESS

CONTROL AND
DYNAMIC SYSTEMS

*Advances in Theory
and Applications*

Volume 79

CONTRIBUTORS TO THIS VOLUME

PETER M. BAINUM
J. P. BARBOT
GIUSEPPE CASALINO
GERALD COOK
M. DJEMAI
ANTONELLA FERRARA
ZHIQIANG GAO
ANDRE GARCIA
ERIC S. HAMBY
TOM T. HARTLEY
MONT HUBBARD
M. JAMSHIDI
YEO-CHOW JUAN
PIERRE T. KABAMBA
RICCARDO MINCIARDI
S. MONACO
D. NORMAND-CYROT
THOMAS PARISINI
ROBERT J. VEILLETTE
GUANG-QIAN XING

CONTROL AND DYNAMIC SYSTEMS

ADVANCES IN THEORY AND APPLICATIONS

Edited by

CORNELIUS T. LEONDES

School of Engineering and Applied Science
University of California, Los Angeles
Los Angeles, California

VOLUME 79: DIGITAL CONTROL SYSTEMS IMPLEMENTATION AND COMPUTATIONAL TECHNIQUES

ACADEMIC PRESS

San Diego New York Boston
London Sydney Tokyo Toronto

Find Us on the Web! http://www.apnet.com

This book is printed on acid-free paper. ♾

Academic Press, Inc.
A Division of Harcourt Brace & Company
525 B Street, Suite 1900, San Diego, California 92101-4495

United Kingdom Edition published by
Academic Press Limited
24-28 Oval Road, London NW1 7DX

International Standard Serial Number: 0090-5267

International Standard Book Number: 0-12-012779-2

PRINTED IN THE UNITED STATES OF AMERICA
96 97 98 99 00 01 QW 9 8 7 6 5 4 3 2 1

CONTENTS

CONTRIBUTORS

Numbers in parentheses indicate the pages on which the authors' contributions begin.

Peter M. Bainum (51), *Department of Mechanical Engineering, Howard University, Washington, D.C. 20059*

J. P. Barbot (203), *Laboratoire des Signaux et Systèmes, CNRS / ESE, 91190 Gif sur Yvette, France*

Giuseppe Casalino (347), *Department of Communications, Computer, and System Sciences, DIST-University of Genoa, 16145 Genova, Italy*

Gerald Cook 117, *Department of Electrical and Computer Engineering, George Mason University, Fairfax, Virginia 22030*

M. Djemai (203), *Laboratoire des Signaux et Systèmes, CNRS / ESE, 91190 Gif sur Yvette, France*

Antonella Ferrara (347), *Department of Communications, Computer, and System Sciences, DIST-University of Genoa, 16145 Genova, Italy*

Zhiqiang Gao 89, *Department of Electrical Engineering, Cleveland State University, Cleveland, Ohio 44115*

Andre Garcia (159), *Hewlett-Packard Company, San Diego, California 92127*

Eric S. Hamby (1), *Aerospace Engineering Department, The University of Michigan, Ann Arbor, Michigan 48109*

Tom T. Hartley (117), *Department of Electrical Engineering, The University of Akron, Akron, Ohio 44325*

Mont Hubbard (159), *Department of Mechanical and Aeronautical Engineering, University of California, Davis, Davis, California 95616*

M. Jamshidi (247), *CAD Laboratory for Intelligent and Robotic Systems, Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, New Mexico 87131*

Yeo-Chow Juan (1), *Ford Motor Company, Dearborn, Michigan 48121*

Pierre T. Kabamba (1), *Aerospace Engineering Department, The University of Michigan, Ann Arbor, Michigan 48109*

Riccardo Minciardi (347), *Department of Communications, Computer, and System Sciences, DIST-University of Genoa, 16145 Genova, Italy*

S. Monaco (203), *Università di Roma "La Sapienza," 00184 Roma, Italy*

D. Normand-Cyrot (203), *Laboratoire des Signaux et Systèmes, CNRS / ESE, 91190 Gif sur Yvette, France*

Thomas Parisini (347), *Department of Electrical, Electronic, and Computer Engineering, DEEI-University of Trieste, 34175 Trieste, Italy*

Robert J. Veillette (117), *Department of Electrical Engineering, The University of Akron, Akron, Ohio 44325*

Guang-Qian Xing (51), *Department of Mechanical Engineering, Howard University, Washington, D.C. 20059 and Beijing Institute of Control Engineering, Beijing People's Republic of China*

PREFACE

Effective control concepts and applications date back over millennia. One very familiar example of this is the windmill. It was designed to derive maximum benefit from windflow, a simple but highly effective optimization technique. Harold Hazen's 1932 paper in the *Journal of the Franklin Institute* was one of the earlier reference points wherein an analytical framework for modern control theory was established. There were many other notable items along the way, including the MIT Radiation Laboratory Series volume on servomechanisms, the Brown and Campbell book, *Principles of Servomechanisms*, and Bode's book, *Network Analysis and Synthesis Techniques*, all published shortly after mid-1945. However, it remained for Kalman's papers of the late 1950s (which established a foundation for modern state-space techniques) and the tremendous evolution of digital computer technology (which was underpinned by the continuous giant advances in integrated electronics) to establish truly powerful control systems techniques for increasingly complex systems. Today we can look forward to a future that is rich in possibilities in many areas of major significance, including manufacturing systems, electric power systems, robotics, aerospace systems, and many other systems with significant economic, safety, cost, and reliability implications. Thus, this volume is devoted to the most timely theme of "Digital Control Systems Implementation and Computational Techniques."

The implementation of digital control systems begins with the conversion of the continuous input to the digital control system to a digital signal. The initial contribution to this volume, "Optimal Hold Functions for Digital Control Functions," by Eric S. Hamby, Yeo-Chow Juan, and Pierre T. Kabamba, is an in-depth treatment of a new approach to this conversion problem which is called the "Generalized Sampled-Data Hold Function Control" (GSHF). In this new GSHF approach, the digital control system treats this conversion process as a design optimization problem. Illustrative examples of this optimal design process are included.

The concept of controllability is one of the cornerstones of modern digital control systems. Yet, despite its fundamental importance from a theoretical point of view, its practical utility for digital control system implementation is limited by its binary nature. That is, a system is either controllable or uncontrollable. There is no provision for consideration of the more subtle question: How controllable is the system? In fact, the desirability of a degree of controllability, which was earlier called the controllability index concept, has been recognized in the literature since 1961. The next contribution, "Actuator Placement Using Degree of Controllability for Discrete-Time Systems," by Guang-Qian Xing and Peter M. Bainum is an in-depth treatment of this subject and the techniques involved.

Reconfigurable control systems possess the ability to accommodate system failures based upon *a priori* assumed conditions. Such systems are also known as restructurable, self-repairing, or failure-tolerant systems. The research in this area was initially motivated by the control systems problems encountered in aircraft control system design in the early 1980s. This is understandable because of the fundamental importance of flight safety. "Techniques in Reconfigurable Control System Design," by Zhiqiang Gao is a rather comprehensive treatment of techniques in this relatively new area in which much further work needs to be done. What makes the reconfigurable control system design problem unique and challenging is that control law redesign must be carried out in real time without human intervention.

In digital control systems for which the system parameters are known, it is possible to algebraically cancel the polynomials representing the system dynamics via feedback compensation to obtain a finite settling time with zero error for a given input. This technique is known as deadbeat control. Separately, another control technique that can be used when system parameters are known involves the tracking of any input signal with a one-step delay between the input and the output. This technique is referred to as one-step-ahead control. The contribution "Techniques in Deadbeat and One-Step-Ahead Control," by Tom T. Hartley, Robert J. Veillette, and Gerald Cook, compares and contrasts deadbeat and one-step-ahead control techniques with the intention of contributing to the unification of the two areas, and to clear some possible points of confusion. An extensive and rather comprehensive retrospective bibliography is included as an invaluable part of this contribution.

Digital control techniques are expanding into many diverse areas of applications of substantial significance. The contribution, "Discrete-Time LQR Techniques in the Control of Modern Canals," by Andre Garcia and Mont Hubbard, is a particularly interesting example of such applications. It also makes the point that the control systems engineer bears a certain amount of responsibility in identifying these expanding areas of potentially important application of the techniques of digital control systems.

In practice, a substantial number of systems can be separated into fast and slow dynamics, and this allows the use of reduction techniques and composite control. These design techniques are referred to as singularly perturbed design techniques in the contribution "Analysis and Control of Nonlinearly Singularly Perturbed Systems under Sampling," by J. P. Barbot,

M. Djemai, S. Monaco, and D. Normand-Cyrot. Digital control system design techniques are presented which are robust with respect to stable, unmodelled high-frequency dynamics. Examples are included which illustrate the effectiveness of the techniques presented.

The single most powerful and widely diversified system for the design and analysis of digital control systems is the MATLAB system. This system has a very wide and continually growing variety of design and synthesis techniques for a wide variety of digital control systems. These many diverse design technique packages for various classes of problems have come to be called toolboxes and are universally utilized on the international scene by many practitioners. The contribution "CAD Techniques in Control Systems," by M. Jamshidi is an in-depth treatment of CAD techniques, in general, and MATLAB, in particular.

In order for digital control system design techniques to be truly effective, both the model for the system under control must be determined to an adequate degree (which is not a trivial aspect in many cases) and an effective or optimal control must be developed for the determined model. The contribution "Implicit Model Techniques and Their Application to LQ Adaptive Control," by Giuseppe Casalino, Antonella Ferrara, Riccardo Minciardi, and Thomas Parisini, treats these issues comprehensively and includes the presentation of highly effective techniques. Simulation results for several examples are presented which clearly manifest the effectiveness of the techniques described.

This volume on digital control systems implementation and computational techniques clearly reveals the significance and power of the techniques available and, with continuing further developments, the essential role they will play in a wide variety of applications. The authors are all to be highly commended for their splendid contributions, which will provide a significant and unique reference for students, research workers, computer scientists, practicing engineers, and others on the international scene for years to come.

This Page Intentionally Left Blank

Optimal Hold Functions for Digital Control Systems

Eric S. Hamby[†]
Yeo-Chow Juan[‡]
Pierre T. Kabamba[†]

† Aerospace Engineering Department
The University of Michigan
Ann Arbor, Michigan 48109-2118

‡ Ford Motor Company
20000 Rotunda Drive
Dearborn, Michigan 48121

I. INTRODUCTION

The classical approach to sampled-data control assumes that the continuous-time control input is generated from the discrete-time output of the digital controller using pre specified digital-to-analog conversion devices -- typically zero-order or first-order hold [1]. Recently, however, a new approach to sampled-data control has been introduced [2-4]. This method is called "Generalized Sampled-Data Hold Function Control" (GSHF), and its original feature is to consider the hold function as a design parameter. GSHF control has been investigated for finite dimensional, continuous time systems [2, 3, 5-7], finite dimensional, discrete time systems [8, 9], and infinite dimensional, continuous time systems [10-13]. In addition, the advantages and disadvantages of GSHF control, compared to classical sampled-data control, have been documented. Roughly speaking, GSHF control appears capable of handling *structured* uncertainties [7, 14], but may provide little robustness in the face of *unstructured* uncertainties [15-17].

CONTROL AND DYNAMICS SYSTEMS, VOL. 79

This paper presents solutions to two classes of optimal design problems in GSHF control: regulation and tracking for an analog plant in a standard 4-block configuration. The distinguishing feature of these optimization problems is that the performance index explicitly penalizes the intersampling behavior of the closed loop system, instead of penalizing only its discrete-time behavior[1]. This is accomplished by using penalty indices that are time integrals of a quadratic function of the state vector and control input of the analog plant. The optimization problem then becomes a standard finite-horizon optimal control problem where the "control input" is the hold function. Existence and uniqueness of an optimal hold function are proven under the assumption of a "fixed monodromy," that is, the discrete-time behavior of the closed-loop system is specified.

The specific problems we solve in regulation (Section II) are the GSHF control versions of the well-known LQ and LQG regulators. These results are a minor generalization of the results in [19] because they are formulated for a plant in a standard 4-block configuration. We also treat the question: "When can we expect the optimal hold function to be close to a zero-order hold?" Specifically, we give a sufficient condition under which the optimal hold function is exactly a zero-order hold, and the answer follows by continuity.

We then consider (Section III) the problem of causing the output of an analog system to asymptotically track a reference signal at the sampling times. This reference signal is itself assumed to be the output of a linear time-invariant system that is observable, but not controllable. After augmenting the dynamics of the plant-controller system with the reference model, we apply the results of Section II. The difficulty is that this augmented system is uncontrollable due to the (possibly unstable) reference model, even though the original plant is

---

[1]Note that Linear-Quadratic Regulation, with an explicit penalty on the intersampling behavior, was treated in [18] for the case of a zero-order hold.

controllable. We must therefore make the dynamics of the reference model unobservable through the regulation error. We give necessary and sufficient conditions for this, which are shown to satisfy the *Internal Model Principle* [20]. We then optimize the hold function with respect to a criterion that reflects the intersample output tracking error and the control energy.

In Section IV, we illustrate several features of optimal GSHF control through examples, by comparison with zero-order hold conversion. We show that optimizing the hold function may yield substantial gain, as measured by the performance index. We identify instances where the optimal hold function can be expected to be close to a zero-order hold. We also show that, sometimes, optimizing the hold function may not yield much improvement, even though the optimal hold function is far from a zero-order hold. Finally, we show that optimal GSHF control can be used to achieve ripple-free deadbeat.

We use the following standard notation: superscript T denotes matrix transpose; $E[\cdot]$ expected value; $tr(\cdot)$ denotes the trace of a matrix; $\delta(\cdot)$ denotes the Kronecker symbol in both the discrete-time and continuous-time case; I_p denotes the identity matrix of order p. Let $X \in \Re^{m\times n}$ and denote the columns of X as x_i, $i = 1,\ldots,n$, i.e., $X = [x_1,\ldots,x_n]$. The *vec* operator on X is defined as $vec(X) = col[x_1, x_2,\ldots,x_n]$, where $vec(X) \in \Re^{mn\times 1}$.

II. OPTIMAL HOLD FUNCTIONS FOR SAMPLED-DATA REGULATION

A. PROBLEM FORMULATION

Consider finite dimensional, linear time invariant, continuous-time systems under sampled-data regulation as follows (see Figure 1):

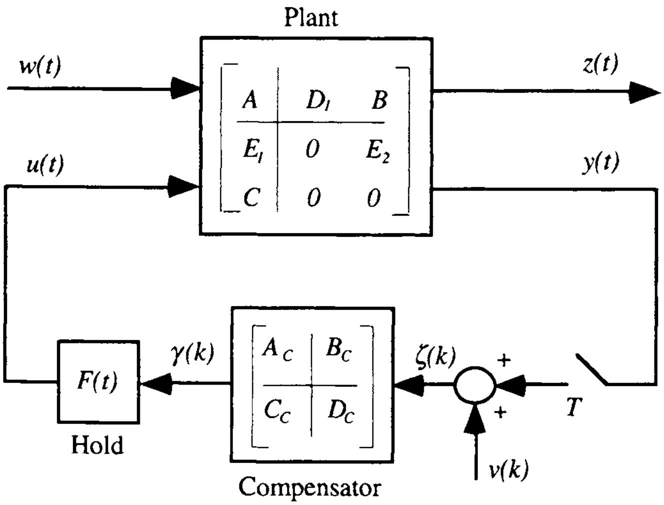


Figure 1 Standard 4-Block Sampled-Data Configuration

Plant and sampler:

$$\dot{x}(t) = Ax(t) + Bu(t) + D_1 w(t), \tag{1}$$

$$\zeta(k) = Cx(kT) + v(k), \tag{2}$$

Digital compensator:

$$x_C(k+1) = A_C x_C(k) + B_C \zeta(k), \tag{3}$$

$$\gamma(k) = C_C x_C(k) + D_C \zeta(k), \tag{4}$$

Digital-to-analog conversion (hold device):

$$u(t) = F(t)\gamma(k), \;\; t \in \left[kT, (k+1)T\right), \tag{5}$$

$$F(t) = F(t+T), \;\; \forall t, \tag{6}$$

Performance:

$$z(t) = E_1 x(t) + E_2 u(t), \tag{7}$$

where $x(t) \in \Re^n$ is the plant state vector; $u(t) \in \Re^m$ is the control input; $w(t) \in \Re^s$ is an exogenous input vector; $\zeta(k) \in \Re^p$ and $v(k) \in \Re^p$ are the discrete measurement vector and discrete measurement noise vector, respectively; $x_C(k) \in \Re^q$ is the compensator state vector; $\gamma(k) \in \Re^r$ is the compensator output; $T > 0$ is the sampling period; $F(t) \in \Re^{m \times r}$ is a T-periodic, integrable, and bounded matrix representing a hold function; $z(t) \in \Re^l$ is a performance vector; and the real matrices A, B, D_1, C, A_C, B_C, C_C, D_C, E_1, and E_2 have appropriate dimensions.

The formalism of Eqs. (1)-(7) is quite general. For instance, zero-order hold (first-order hold, ith-order hold) control is obtained by letting the hold function $F(t)$ be a constant (first-degree polynomial, ith-degree polynomial, respectively).

For a given hold function $F(t)$, $t \in [0, T)$, the problem of designing the matrices A_C, B_C, C_C, and D_C for performance of the corresponding discrete-time system has been extensively treated in the literature [1]. Our objective in this paper is, for a given compensator Eqs. (3)-(4), to determine time histories of the hold function $F(t)$ in Eqs. (5)-(6) that will minimize an H_2 performance criterion associated with the sampled-data system given in Eqs. (1)-(6):

Upon loop closure, the plant state and control between samples satisfy

$$\begin{aligned} x(kT+t) = \Phi(t)x(kt) + D(t)C_C x_C(k) + D(t)D_C v(k) + \\ \omega(kT+t), \qquad t \in [0, T), \end{aligned} \tag{8}$$

$$\begin{aligned} u(kT+t) = F(t)D_C C x(kT) + F(t)C_C x_C(k) + \\ F(t)D_C v(k), \quad t \in [0, T), \end{aligned} \tag{9}$$

where

$$D(t) := \int_0^t e^{A(t-\sigma)} B F(\sigma) d\sigma, \qquad t \in [0, T), \tag{10}$$

$$\Phi(t) := e^{At} + D(t)D_C C, \quad t \in [0,T), \tag{11}$$

$$\omega(kT+t) := \int_{kT}^{kT+t} e^{A(kT+t-\sigma)} D_1 w(\sigma) d\sigma, \quad t \in [0,T). \tag{12}$$

Defining

$$x_a(k) := \left[x^T(kT) \quad x_C^T(kT)\right]^T \in \Re^{n+q}, \tag{13}$$

$$\Psi_a := \begin{bmatrix} e^{AT} + D(T)D_C C & D(T)C_C \\ B_C C & A_C \end{bmatrix} \in \Re^{(n+q)\times(n+q)}, \tag{14}$$

$$D_a := \begin{bmatrix} D(T)D_C \\ B_C \end{bmatrix} \in \Re^{(n+q)\times p}, \tag{15}$$

$$\omega_a(k) := \begin{bmatrix} \omega[(k+1)T] \\ 0 \end{bmatrix} \in \Re^{n+q}, \tag{16}$$

then the closed loop equations for the discrete-time system are

$$x_a(k+1) = \Psi_a x_a(k) + D_a v(k) + \omega_a(k). \tag{17}$$

The closed loop monodromy matrix is defined as Ψ_a in Eqs. (14) and denotes the state transition matrix of the regulated discrete-time system over one period.

Definition 2.1: The design problem of finding an optimal hold function $F(t)$, $t \in [0,T)$ is called a *fixed monodromy* (*free monodromy*) problem if *D(T)* in Eq. (10) is specified (not specified).

A fixed monodromy problem must therefore satisfy a design constraint of the form

$$D(T) = G, \tag{18}$$

where typically, the matrix G is chosen such that the closed loop monodromy matrix Ψ_a of Eq. (14) defines a stable discrete-time system (Eq. (17)).

B. LINEAR QUADRATIC GAUSSIAN REGULATION

Throughout this section we assume that $w(t)$ and $v(k)$ of Eqs. (1) and (2) are stationary Gaussian processes satisfying

$$E\left[w(t)\right]=0, \qquad t\in\Re, \tag{19}$$

$$E\left[v(k)\right]=0, \qquad k\in\mathbf{Z}, \tag{20}$$

$$E\left[w(t)v^T(k)\right]=0, \qquad t\in\Re,\ k\in\mathbf{Z}, \tag{21}$$

$$E\left[w(t)w^T(\tau)\right]=R_w\delta(t-\tau), \qquad t,\tau\in\Re,\ R_w\geq 0, \tag{22}$$

$$E\left[v(k)v^T(\iota)\right]=R_v\delta(k-\iota), \qquad k,\iota\in\mathbf{Z},\ R_v\geq 0, \tag{23}$$

$$E\left[w(t)x^T(0)\right]=0, \qquad t\in\Re, \tag{24}$$

$$E\left[v(k)x^T(0)\right]=0, \qquad k\in\mathbf{Z}. \tag{25}$$

Equation (12) implies that $\omega(k)$ is a stationary, zero-mean Gaussian sequence with covariance kernel

$$E\left[\omega(k)\omega^T(\iota)\right]=R_\omega(T)\delta(k-\iota), \tag{26}$$

where

$$R_\omega(T)=\int_0^T e^{A(T-\sigma)}D_1R_wD_1^Te^{A^T(T-\sigma)}d\sigma. \tag{27}$$

The performance index is defined as

$$J = \lim_{t_f \to \infty} E\left\{ \frac{1}{t_f} \int_0^{t_f} z^T(t) z(t) dt \right\}. \tag{28}$$

The criterion, Eq. (28) can also be computed as shown below.

Proposition 2.1: Suppose a hold function $F(t)$, $t \in [0,T)$ stabilizes asymptotically Eq. (17). Then the criterion in Eq. (28) has the form

$$J = \lim_{k \to \infty} E\left\{ \frac{1}{T} \int_{kT}^{(k+1)T} z^T(t) z(t) dt \right\}. \tag{29}$$

Proof: See Appendix.

Proposition 2.2: Suppose a hold function $F(t)$, $t \in [0,T)$ stabilizes asymptotically Eq. (17). Then the criterion in Eq. (28) can be computed as follows:

$$J = \frac{1}{T} tr\left\{ LM(T) + E_1^T E_1 P(T) + R_v N(T) \right\}, \tag{30}$$

where $L \in \Re^{(n+q)\times(n+q)}$, $M(t) \in \Re^{(n+q)\times(n+q)}$, $P(t) \in \Re^{n\times n}$, and $N(t) \in \Re^{p\times p}$ are computed as follows:

$$\dot{D}(t) = AD(t) + BF(t); \quad D(0) = 0, \tag{31}$$

$$\dot{M}(t) = \begin{bmatrix} e^{A^T t} + C^T D_C^T D^T(t) \\ C_C^T D^T(t) \end{bmatrix} E_1^T E_1 \begin{bmatrix} e^{At} + D(t) D_C C & D(t) C_C \end{bmatrix} + \tag{32}$$
$$\begin{bmatrix} e^{A^T t} + C^T D_C^T D^T(t) \\ C_C^T D^T(t) \end{bmatrix} E_1^T E_2 \begin{bmatrix} F(t) D_C C & F(t) C_C \end{bmatrix} +$$

$$\begin{bmatrix} C^T D_C^T F^T(t) \\ C_C^T F^T(t) \end{bmatrix} E_2^T E_1 \begin{bmatrix} e^{At} + D(t) D_C C & D(t) C_C \end{bmatrix} +$$
$$\begin{bmatrix} C^T D_C^T F^T(t) \\ C_C^T F^T(t) \end{bmatrix} E_2^T E_2 \begin{bmatrix} F(t) D_C C & F(t) C_C \end{bmatrix}; \quad M(0) = 0,$$

$$\begin{aligned} \dot{N}(t) = {} & D_C^T D^T(t) E_1^T E_1 D(t) D_C + D_C^T F^T(t) E_2^T E_2 F(t) D_C + \\ & D_C^T F^T(t) E_2^T E_1 D(t) D_C + D_C^T D^T(t) E_1^T E_2 F(t) D_C; \quad N(0) = 0, \end{aligned} \tag{33}$$

$$\dot{P}(t) = R_\omega(t); \quad P(0) = 0, \tag{34}$$

$$\dot{R}_\omega(t) = A R_\omega(t) + R_\omega(t) A^T + D_1 R_w D_1^T; \quad R_\omega(0) = 0, \tag{35}$$

$$\begin{aligned} & \begin{bmatrix} e^{AT} + D(T) D_C C & D(T) C_C \\ B_C C & A_C \end{bmatrix} L \begin{bmatrix} e^{A^T T} + C^T D_C^T D^T(T) & C^T B_C^T \\ C_C^T D^T(T) & A_C^T \end{bmatrix} + \\ & \begin{bmatrix} D(T) D_C \\ B_C \end{bmatrix} R_v \begin{bmatrix} D_C^T D^T(T) & B_C^T \end{bmatrix} + \begin{bmatrix} R_\omega(T) & 0 \\ 0 & 0 \end{bmatrix} - L = 0. \end{aligned} \tag{36}$$

Proof: See Appendix.

1. FREE MONODROMY

Proposition 2.3: A hold function $F(t)$, $t \in [0, T)$ which stabilizes Eq. (17) asymptotically and minimizes Eq. (28) must satisfy:

$$\begin{aligned} & B^T \Theta(t) - 2 E_2^T E_1 \begin{bmatrix} e^{At} + D(t) D_C C & D(t) C_C \end{bmatrix} L \begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} - \\ & 2 E_2^T E_1 D(t) D_C R_v D_C^T - 2 E_2^T E_2 F(t) Q = 0, \end{aligned} \tag{37}$$

where $\Theta(t) \in \Re^{n \times r}$ satisfies

$$\dot{\Theta}(t) = -A^T\Theta(t) + 2E_1^TE_1\left[e^{At} + D(t)D_CC \quad D(t)C_C\right]L\begin{bmatrix} C^TD_C^T \\ C_C^T \end{bmatrix} + \tag{38}$$

$$2E_1^TE_1D(t)D_CR_vD_C^T + 2E_1^TE_2F(t)Q,$$

$$\Theta(T) = -2\left[I_n \quad 0\right]K\left\{\begin{bmatrix} e^{AT} + D(T)D_CC & D(T)C_C \\ B_CC & A_C \end{bmatrix}L\begin{bmatrix} C^TD_C^T \\ C_C^T \end{bmatrix} + \tag{39}$$

$$\begin{bmatrix} D(T)D_C \\ B_C \end{bmatrix}R_vD_C^T\right\},$$

$$\begin{bmatrix} e^{A^TT} + C^TD_C^TD^T(T) & C^TB_C^T \\ C_C^TD^T(T) & A_C^T \end{bmatrix}K\begin{bmatrix} e^{AT} + D(T)D_CC & D(T)C_C \\ B_CC & A_C \end{bmatrix} + \tag{40}$$

$$M(T) - K = 0,$$

with

$$Q := \left[D_CC \quad C_C\right]L\begin{bmatrix} C^TD_C^T \\ C_C^T \end{bmatrix} + D_CR_vD_C^T, \tag{41}$$

and the matrices $D(t)$, $M(t)$, and L are given by Eqs. (31), (32), and (36), respectively.

Proof: See Appendix.

2. FIXED MONODROMY

Proposition 2.4: If the triple (A, B, C) is minimal, and the matrix G of Eq. (18) is such that the system in Eq. (17) is asymptotically stable, then for almost all T, the hold function $F(t)$, $t \in [0, T)$ which minimizes Eq. (28) subject to Eqs. (1)-(7) and (18)-(27) exists and is unique. It satisfies the following:

$$\dot{D}(t) = AD(t) + B\left(E_2^T E_2\right)^{-1}\left\{\frac{1}{2}B^T\Theta(t) - E_2^T E_1\left(\left[e^{At} + D(t)D_C C \quad D(t)C_C\right]L\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + D(t)D_C R_v D_C^T\right)\right\}Q^{-1}; \qquad D(0) = 0,\ D(T) = G, \tag{42}$$

$$\dot{\Theta}(t) = -A^T\Theta(t) + 2E_1^T E_1\left[e^{At} + D(t)D_C C \quad D(t)C_C\right]L\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + 2E_1^T E_1 D(t)D_C R_v D_C^T + 2E_1^T E_2\left(E_2^T E_2\right)^{-1}\left\{\frac{1}{2}B^T\Theta(t) - E_2^T E_1\left(\left[e^{At} + D(t)D_C C \quad D(t)C_C\right]L\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + D(t)D_C R_v D_C^T\right)\right\}, \tag{43}$$

$$F(t) = \left(E_2^T E_2\right)^{-1}\left\{\frac{1}{2}B^T\Theta(t) - E_2^T E_1\left(\left[e^{At} + D(t)D_C C \quad D(t)C_C\right] \times L\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + D(t)D_C R_v D_C^T\right)\right\}Q^{-1}, \tag{44}$$

and L is the positive semidefinite solution of Eq. (36).

Proof: See Appendix

In sampled-data control design, the discrete-time plant model is typically generated using a prespecified hold device such as a zero-order hold [1]. Our experience has shown that under certain conditions the optimal GSHF is close to a zero-order hold. This motivates the next proposition which gives a sufficient condition for the optimal GSHF to be a zero-order hold.

Proposition 2.5: Consider the fixed monodromy case for $m = r$ and assume the columns of B are linearly independent. Let *D(T)* be determined by requiring that the discretized plant with GSHF be the same as the discretized plant with zero-order hold (ZOH). Finally, let the plant state dynamics and the state weighting matrices be parameterized as εA and εE_1, where $\varepsilon \geq 0$. Then, as $\varepsilon \to 0$ the optimal GSHF tends to a ZOH.

Proof: See Appendix.

C. LINEAR QUADRATIC REGULATION

Throughout this section we assume there is no disturbance and no measurement noise: we are regulating the transient behavior of the closed loop system against nonzero initial conditions. For Eqs. (1)-(3) we now assume the following:

$$w(t) = 0, \qquad v(k) = 0, \tag{45}$$

$$E[x(0)] = 0, \qquad E[x_C(0)] = 0, \tag{46}$$

$$E\left[\begin{bmatrix} x(0) \\ x_C(0) \end{bmatrix}\begin{bmatrix} x^T(0) & x_C^T(0) \end{bmatrix}\right] = X_0. \tag{47}$$

The performance index has the form

$$J = E\left(\int_0^\infty z^T(t)z(t)dt\right). \tag{48}$$

1. FREE MONODROMY

Proposition 2.6: A hold function $F(t)$, $t \in [0, T)$ which asymptotically stabilizes Eq. (17) and minimizes Eq. (48) subject to Eqs. (1)-(7) and Eqs. (45)-(48) must satisfy Eqs. (31), (32), and (36)-(40) where X_0 replaces

$$\begin{bmatrix} R_\omega(T) & 0 \\ 0 & 0 \end{bmatrix},$$

and $R_v = 0$. The optimal value of Eq. (48) is then

$$J = tr\left(KX_0\right). \tag{49}$$

Proof: See Appendix.

2. FIXED MONODROMY

Proposition 2.7: If the triple (A, B, C) is minimal, and the matrix G of Eq. (18) is such that the system in Eq. (17) is asymptotically stable, then for almost all T, the hold function $F(t)$, $t \in [0, T)$ which minimizes Eq. (48) subject to Eqs. (1)-(7), Eqs. (45)-(48), and Eq. (18) exists and is unique. It satisfies Eqs. (42)-(44) where X_0 replaces

$$\begin{bmatrix} R_\omega(T) & 0 \\ 0 & 0 \end{bmatrix},$$

and $R_v = 0$. The optimal value of Eq. (48) is then given by Eq. (49).

Proof: See Appendix.

Remark 2.1: Propositions 2.6 and 2.7 reveal that the linear quadratic regulator is a particular case of the linear quadratic Gaussian regulator where X_0 replaces

$$\begin{bmatrix} R_\omega(T) & 0 \\ 0 & 0 \end{bmatrix},$$

and $R_v = 0$. In other words, optimizing the transient performance of Eq. (48) for a noiseless undisturbed system is equivalent to optimizing the steady state performance (Eq. (28)) of the same system under some properly defined perturbations.

Remark 2.2: Propositions 2.3-2.4 and 2.6-2.7 illustrate the fact that fixed monodromy problems are easier to solve than free monodromy problems. Not only can we guarantee existence and uniqueness of an optimal fixed monodromy hold function, but it can be computed directly at the expense of solving linear equations. On the other hand, for free monodromy problems we cannot guarantee either existence or uniqueness of a solution, and we cannot guarantee that an iterative algorithm will always converge. The hierarchy of difficulty between fixed and free monodromy problems is reminiscent of the problem of L_2 optimal model reduction [21] where, if the poles of the optimal reduced order model are free, it is not guaranteed to exist, to be unique, or to be computable by a convergent algorithm; whereas if these poles are fixed, the reduced order model is computed directly be solving linear equations.

III. OPTIMAL DISCRETE TIME TRACKING

A. PROBLEM FORMULATION

Consider finite dimensional, linear, time invariant, continuous-time systems under discrete-time tracking as shown in Figure 2. The model of the plant and sampler are as defined in Eqs. (1) and (2) with $D_1 = 0$ and $v(k) = 0$, and the digital compensator is defined in Eqs. (3) and (4).

The reference variable $y_r(t)$ is modeled as the output of a linear, time invariant, continuous-time system as follows:

$$\dot{x}_r(t) = A_r x_r(t), \tag{50}$$

$$y_r(t) = C_r x_r(t), \tag{51}$$

where $x_r(t) \in \Re^{\beta}$ is the reference model state vector, $y_r(t) \in \Re^{p}$ is the output vector of the reference model, $\gamma_r(k) \in \Re^{p}$ is the discrete output of the reference model, and the matrices A_r and C_r are real and have appropriate dimension. The reference model described by Eqs. (50) and (51) is uncontrollable but will be assumed observable. The usual references such as a step, ramp, and sinusoid fall into this formulation.

The digital-to-analog conversion (hold device) is now defined as

$$u(t) := F(t)\gamma(k) + F_r(t)\gamma_r(k),\ \ t \in [kT, (k+1)T), \tag{52}$$

$$[F(t+T), F_r(t+T)] = [F(t), F_r(t)],\ \ \forall t, \tag{53}$$

where $F(t) \in \Re^{m \times r}$ and $F_r(t) \in \Re^{m \times p}$ are T-periodic, integrable, and bounded matrices representing hold functions.

The output tracking error, $e(t)$ is defined as

$$e(t) := y(t) - y_r(t) = Cx(t) - C_r x_r(t). \tag{54}$$

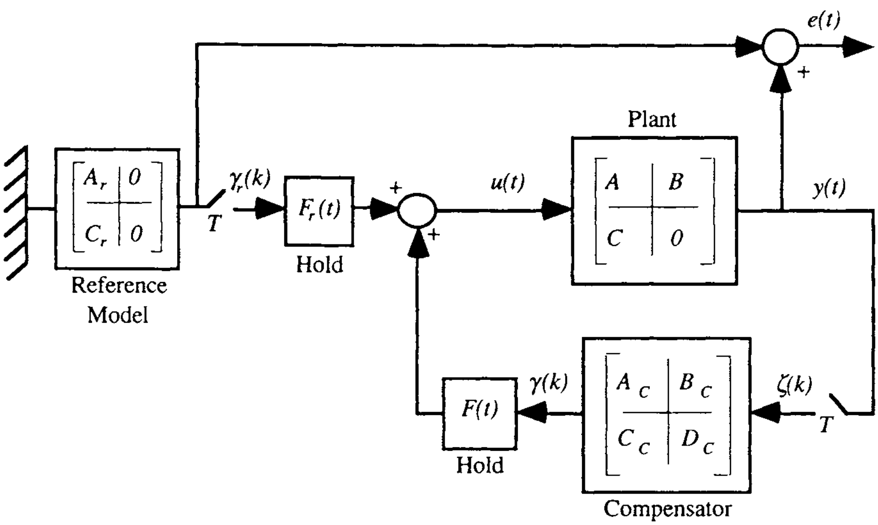


Figure 2 Configuration for Tracking by Generalized Sampled Data Hold Function Control

Upon loop closure, the augmented state $x_a(k) := \left[x^T(kT) \quad x_C^T(k) \quad x_r^T(kT)\right]^T$ between sampling instants is described by

$$x_a(kT+t) = \Psi_a(t)x_a(k), \qquad t \in [0,T), \tag{55}$$

where

$$\Psi_a(t) := \begin{bmatrix} e^{At} + D(t)D_C C & D(t)C_C & D_r(t)C_r \\ B_C C & A_C & 0 \\ 0 & 0 & e^{A_r t} \end{bmatrix}, \quad t \in [0,T), \tag{56}$$

$$D_r(t) := \int_0^t e^{A(t-\sigma)}BF_r(\sigma)d\sigma, \quad t \in [0,T), \tag{57}$$

and $D(t)$ is defined in Eq. (10). Assume that $D_r(T)$ is given as

$$D_r(T) = G_r, \tag{58}$$

and $D(T)$ is as given in Eq. (18).

Furthermore, define

$$\Phi(T,0) := e^{AT}, \tag{59}$$

$$\Phi_r(T,0) := e^{A_r T}. \tag{60}$$

Then, the closed loop monodromy matrix of the augmented system is

$$\Psi_a(T) = \begin{bmatrix} \Phi(T,0) + GD_C C & GC_C & G_r C_r \\ B_C C & A_C & 0 \\ 0 & 0 & \Phi_r(T,0) \end{bmatrix}. \tag{61}$$

Our objective in this section is to determine the time histories of the hold functions $F(t)$ and $F_r(t)$ which will suppress the dynamics of the reference model from the output tracking error at the sampling times and will optimize a performance index related to intersample error and intersample control energy.

Throughout this section, we make the following assumptions:

1. The pair $(\Phi(T,0), C)$ is observable.
2. The matrices $\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}$ and $\Phi_r(T,0)$ have no common eigenvalues.

These assumptions imply no loss of generality because the pair $(\Phi(T,0), C)$ will be observable for almost all T if the pair (A,C) is observable, and we typically want an asymptotically stable system to track unstable or marginally stable signals.

B. DISCRETE TIME TRACKING

In this subsection, a necessary and sufficient condition for suppressing the dynamics of $\Phi_r(T,0)$ from the output tracking error at the sampling times is derived. A particular case which assumes that A_r and $\Phi_r(T,0)$ have distinct eigenvalues is also considered. We first choose the matrix G and a realization of the compensator, (A_C, B_C, C_C, D_C), so that $\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}$ defines a stable discrete time system. The matrix G_r is then determined from the following:

Proposition 3.1: A necessary and sufficient condition for the modes of $\Phi_r(T,0)$ to be suppressed in the sequence $e(kT)$ of Eq. (54) is that the matrix G_r satisfy

$$\begin{bmatrix} I_\beta \otimes (\Phi(T,0)+GD_C C) - \Phi_r^T(T,0)\otimes I_n & I_\beta \otimes GC_C & C_r^T \otimes I_n \\ I_\beta \otimes B_C C & I_\beta \otimes A_C - \Phi_r^T(T,0)\otimes I_q & 0 \\ I_\beta \otimes C & 0 & 0 \end{bmatrix} \times$$

$$\begin{bmatrix} vecK \\ vecK_C \\ vecG_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ vecC_r \end{bmatrix}. \tag{62}$$

Proof: See Appendix.

When Eq. (62) is satisfied, the only modes appearing in the sequence *e(kT)* are those of $\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}$. Both G and (A_C, B_C, C_C, D_C) have been chosen so that $\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}$ is asymptotically stable. Therefore, we will have $\lim_{k\to\infty} e(kT) = 0$ *i.e.* asymptotic tracking.

Remark 3.1: Equation (62) contains $\beta(n+q+p)$ equations and $\beta n+\beta q+pn$ unknowns, K, K_C, and G_r. The "degree of indeterminacy" is $p(n-\beta)$ which indicates that tracking is in general impossible if $\beta > n$ *i.e.* if we attempt to track a class of reference signals that is "too rich."

Remark 3.2: Equation (62) can be solved for K, K_C, and G_r iff the vector on the right hand side belongs to the range space of the matrix on the left hand side.

Remark 3.3: If the pair (A,B) in Eq. (1) is controllable, given G and G_r, it is always possible to solve Eqs. (10) and (57) for *F(t)* and $F_r(t)$.

If, as a particular case, we assume that A_r and $\Phi_r(T,0)$ have distinct eigenvalues, then a more transparent necessary condition than Eq. (62) is derived. This condition is the GSHF version of the Internal Model Principle.

Proposition 3.2: Suppose A_r and $\Phi_r(T,0)$ have distinct eigenvalues. A necessary condition for the modes of $\Phi_r(T,0)$ to be suppressed in the sequence *e(kT)* of Eq. (54) is that spectrum of the matrix $\begin{bmatrix} \Phi(T,0)+GD_C C+G_r C & GC_C \\ B_C C & A_C \end{bmatrix}$ contain that of $\Phi_r(T,0)$.

Proof: See Appendix.

Remark 3.4: For multi-output systems, $p \neq 1$, the condition in Proposition 3.2 is necessary for suppressing the dynamics of the reference model from the error signal, but it is not sufficient.

Proposition 3.3: Suppose A_r and $\Phi_r(T,0)$ have distinct eigenvalues, and we are interested in tracking a single signal, i.e. $p=1$. Then, a necessary and sufficient condition for the modes of $\Phi_r(T,0)$ to be suppressed in the sequence *e(kT)* of Eq. (54) is that spectrum of the matrix $\begin{bmatrix} \Phi(T,0)+GD_C C+G_r C & GC_C \\ B_C C & A_C \end{bmatrix}$ contain that of $\Phi_r(T,0)$.

Proof: See Appendix.

C. OPTIMAL HOLD FUNCTIONS FOR INTERSAMPLING BEHAVIOR

For a given G in Eq. (18) and a given G_r in Eq. (58), there exists an infinite number of $F(t)$ and $F_r(t)$ satisfying Eqs. (10) and (57), respectively. As a result, we want to choose hold functions $F(t)$ and $F_r(t)$ which not only suppress the modes of $\Phi_r(T,0)$ from the sequence *e(kT)* in Eq. (54) but will also optimize a performance index related to intersample error and intersample control energy.

Define

$$\tilde{x}_a(k) := \begin{bmatrix} x^T(kT) & x_r^T(kT) & x_C^T(k) \end{bmatrix}^T, \tag{63}$$

$$\tilde{A}(t) := \begin{bmatrix} e^{At} & 0_{n\times\beta} \end{bmatrix}, \tag{64}$$

$$\tilde{A}_r(t) := \begin{bmatrix} 0_{\beta\times n} & e^{A_r t} \end{bmatrix}, \tag{65}$$

$$\tilde{C} := \begin{bmatrix} C & 0 \\ 0 & C_r \end{bmatrix}, \tag{66}$$

$$\tilde{C}_C := \begin{bmatrix} C_C \\ 0_{p\times q} \end{bmatrix}, \tag{67}$$

$$\tilde{D}_C := \begin{bmatrix} D_C & 0 \\ 0 & I_p \end{bmatrix}, \tag{68}$$

$$\tilde{D}(t) := \begin{bmatrix} D(t) & D_r(t) \end{bmatrix}, \tag{69}$$

$$\tilde{D}(T) := \tilde{G} = \begin{bmatrix} G & G_r \end{bmatrix}, \tag{70}$$

$$\tilde{F}(t) := \begin{bmatrix} F(t) & F_r(t) \end{bmatrix}. \tag{71}$$

Using the above definitions, the output tracking error and control between samples satisfies

$$e(kT+t) = \tilde{E}_1 X(t)\tilde{x}_a(k), \quad t \in [0,T), \tag{72}$$

$$u(kT+t) = U(t)\tilde{x}_a(k), \qquad t \in [0,T), \tag{73}$$

where

$$\tilde{E}_1 := \begin{bmatrix} C & -C_r \end{bmatrix}, \tag{74}$$

$$X(t) := \begin{bmatrix} \tilde{A}(t) + \tilde{D}(t)\tilde{D}_C\tilde{C} & \tilde{D}(t)\tilde{C}_C \\ \tilde{A}_r(t) & 0 \end{bmatrix}, \tag{75}$$

$$U(t) := \tilde{F}(t)\left[\tilde{D}_C\tilde{C} \quad \tilde{C}_C\right]. \tag{76}$$

Proposition 3.4: Consider the criterion

$$J = tr\left\{\int_0^T \left[X^T(t)\tilde{E}_1^T\tilde{E}_1X(t) + U^T(t)\tilde{E}_2^T\tilde{E}_2U(t)\right]dt\right\}, \tag{77}$$

where $\tilde{E}_2$ is a real matrix with rank m. If (A,B) is controllable, then for almost all T there exists a unique hold function $\tilde{F}(t)$, $t \in [0,T)$ which minimizes the cost in Eq. (77) subject to Eqs. (10), (18), (57), and (58). It satisfies the following:

$$\dot{\tilde{D}}(t) = A\tilde{D}(t) - \frac{1}{2}B\left(\tilde{E}_2^T\tilde{E}_2\right)^{-1}B^T\tilde{\Theta}(t)\left(\left[\tilde{D}_C\tilde{C} \quad \tilde{C}_C\right]\begin{bmatrix}\tilde{C}^T\tilde{D}_C^T \\ \tilde{C}_C^T\end{bmatrix}\right)^{-1}, \tag{78}$$

$$\tilde{D}(0) = 0, \quad \tilde{D}(T) = \tilde{G}, \tag{79}$$

$$\dot{\tilde{\Theta}}(t) = -A^T\tilde{\Theta}(t) - 2\left[C^TC \quad -C^TC_r\right]\times \\ \begin{bmatrix}\tilde{A}(t) + \tilde{D}(t)\tilde{D}_C\tilde{C} & \tilde{D}(t)\tilde{C}_C \\ \tilde{A}_r(t) & 0\end{bmatrix}\begin{bmatrix}\tilde{C}^T\tilde{D}_C^T \\ \tilde{C}_C^T\end{bmatrix}, \tag{80}$$

$$\tilde{F}(t) = -\frac{1}{2}\left(\tilde{E}_2^T\tilde{E}_2\right)^{-1}B^T\tilde{\Theta}(t)\left(\left[\tilde{D}_C\tilde{C} \quad \tilde{C}_C\right]\begin{bmatrix}\tilde{C}^T\tilde{D}_C^T \\ \tilde{C}_C^T\end{bmatrix}\right)^{-1}. \tag{81}$$

Proof: Follows from Proposition 2.4.

Remark 3.5: In practice, the fixed monodromy condition in Eq. (79) is specified by choosing the matrix G such that $\begin{bmatrix}\Phi(T,0) + GD_CC & GC_C \\ B_CC & A_C\end{bmatrix}$ defines an asymptotically stable discrete-time system and by choosing the matrix G_r such that Eq. (62) is satisfied.

The idea of Proposition 3.4 is to reformulate the optimal design problem for GSHF control as a standard linear quadratic optimal control problem. A unique optimal hold function exists and can be obtained by solving a two point boundary value problem with Hamiltonian structure.

IV. EXAMPLES

A. REGULATION

In this section, we illustrate the use of the results of Section II. First, we compare two sampled-data regulators for a given plant. The first regulator is obtained using a ZOH preceded by a discrete compensator based on pole assignment. The second regulator is obtained by using the same discrete compensator and optimizing the hold function with respect to the performance index under the "fixed monodromy constraint" that *the discretized plant in the ZOH configuration is the same as the discretized plant in the GSHF configuration*. In other words, we show that it is possible to improve the performance of a feedback loop by adjusting the hold function without changing either the discrete compensator or the discrete model of the plant. Such an improvement is then due exclusively to better intersampling behavior. Note that the discrete compensator has not been chosen for optimality with respect to the performance index, because this optimization would depend on the hold function, and the effect of optimizing the hold function alone would then be difficult to assess. Next, we parameterize the plant dynamics and the state weighting of the second regulator by a parameter, ε and show that as $\varepsilon \to 0$ the optimal GSHF does indeed approach a ZOH. Finally, we show that optimizing the hold function alone does not always yield a significant improvement in the performance index when compared with a ZOH. To do this, we once again compare the performance of two sampled-data regulators for a given plant as described above.

1. HARMONIC OSCILLATOR [19]

The plant is a simple harmonic oscillator with position measurement of the form

$$\dot{x} = \begin{bmatrix} 0 & -5 \\ 5 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix} x. \tag{82}$$

The sampling period is $T = 1$. Note that the period of the plant dynamics is 1.26.

The first regulator is implemented using a dynamic compensator and a ZOH

$$u(t) = \gamma(k), \qquad t \in [kT, (k+1)T). \tag{83}$$

The discretized model of the plant is then

$$x(k+1) = \begin{bmatrix} 0.2837 & 0.9589 \\ -0.9589 & 0.2837 \end{bmatrix} x(k) + \begin{bmatrix} -0.1433 \\ -0.1918 \end{bmatrix} \gamma(k), \qquad \zeta(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k). \tag{84}$$

The dynamic compensator is

$$x_C(k+1) = \begin{bmatrix} -0.0673 & 0.3735 \\ 0.2057 & -0.5000 \end{bmatrix} x_C(k) + \begin{bmatrix} 0.5673 \\ -0.875 \end{bmatrix} \zeta(k), \qquad \gamma(k) = \begin{bmatrix} -1.5100 & 4.0862 \end{bmatrix} x_C(k). \tag{85}$$

and has been chosen so that the closed loop poles are located at $\pm 1.0 \times 10^{-3}$ and $\pm j \times 10^{-3}$. The cost function has the form of Eqs. (47) and (48) with

$$E_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \text{and} \quad X_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{86}$$

For this regulator, the performance index is computed using Proposition 2.7 which yields

$$J_1 = 47.82. \tag{87}$$

The second regulator consists of a sampler with GSHF along with the discrete compensator given in Eq. (85). The hold function is constrained such that the discretized plant is the same as Eq. (84). In other words, the closed loop systems of both regulators have the same dynamics at sampling instants. However, now the hold function is chosen so as to optimize the performance index in Eq. (48). Applying Proposition 2.7 yields the hold function of Figure 3 ($\varepsilon = 1$) and a cost of

$$J_2 = 9.50. \tag{88}$$

which represents an 80% improvement over Eq. (87).

To illustrate Proposition 2.5, we first parameterize the plant dynamics of Eq. (82) and the state weighting of Eq. (86) as εA and εE_1, respectively. Then, for a given ε a discrete compensator is determined according to pole assignment as described above. Finally, the hold function is chosen so as to optimize the performance index in Eq. (48) and is calculated using Proposition 2.7. As shown in Figure 3, the optimal GSHF approaches the ZOH as $\varepsilon \to 0$.

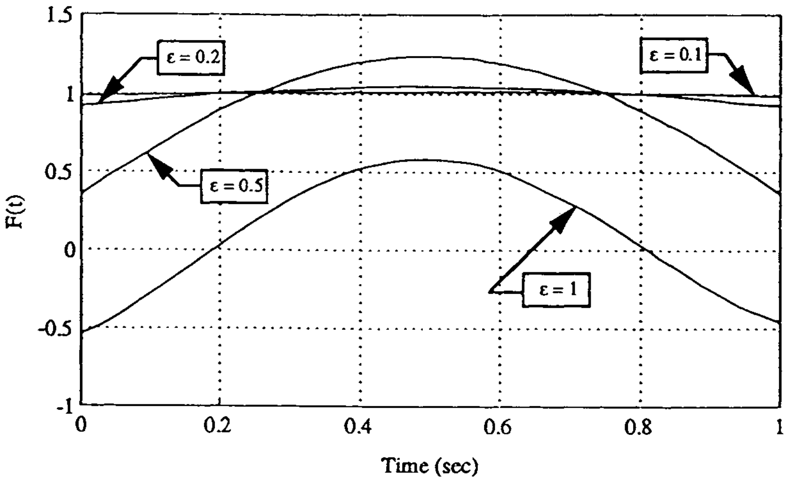


Figure 3 Generalized sampled-data hold functions for harmonic oscillator example

2. FLEXIBLE SYSTEM

The plant is a double mass-spring-damper system with velocity measurement as follows

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -62.5 & 25 & -1.5 & 1 \\ 50 & -50 & 2 & -2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 0 \end{bmatrix} u, \tag{89}$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} x.$$

The sampling period is $T = 1$.

The first regulator is implemented using a dynamic compensator and a ZOH (Eq. (83)). The discretized model of the plant is

$$x(k+1)=\begin{bmatrix}-0.1394 & -0.0172 & -0.0744 & -0.0620\\ 0.0883 & -0.2707 & -0.1241 & -0.1047\\ 1.5501 & 1.2407 & -0.1519 & 0.0325\\ 2.5158 & 2.1361 & 0.0649 & -0.1853\end{bmatrix}x(k)+\begin{bmatrix}0.0154\\ 0.0158\\ -0.0372\\ -0.0620\end{bmatrix}\gamma(k), \tag{90}$$

$$\zeta(k)=\begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}x(k).$$

The dynamic compensator is

$$x_C(k+1)=\begin{bmatrix}-0.1328 & -0.0104 & -0.0658 & -0.0612\\ 0.0952 & -0.2637 & 0.1047 & -0.1039\\ 1.5339 & 1.2243 & 0.7124 & 0.0305\\ 2.4889 & 2.1087 & 1.1793 & -0.1886\end{bmatrix}x_C(k)+\begin{bmatrix}-0.0074\\ -0.2274\\ -0.8673\\ -1.1194\end{bmatrix}\zeta(k), \tag{91}$$

$$\gamma(k)=\begin{bmatrix}0.4331 & 0.4422 & 0.0803 & 0.0538\end{bmatrix}x_C(k).$$

The performance index has the form of Eqs. (47) and (48) with

$$E_1=\begin{bmatrix}I_4\\ 0_{1\times 4}\end{bmatrix},\quad E_2=\begin{bmatrix}0_{4\times 1}\\ 1\end{bmatrix},\quad \text{and}\quad X_0=\begin{bmatrix}I_4 & 0_{4\times 4}\\ 0_{4\times 4} & 0_{4\times 4}\end{bmatrix}. \tag{92}$$

For this regulator, the performance index is computed using Proposition 2.7 which yields

$$J_1=48.83. \tag{93}$$

The second regulator consists of a sampler with GSHF along with the discrete compensator given in Eq. (91). The hold function is constrained as described in the previous example. Applying Proposition 2.7 yields the hold function shown in Figure 4 and a cost of

$$J_2 = 48.67. \tag{94}$$

Note that while the optimal GSHF shown in Figure 4 is considerably different than a ZOH, the improved performance that it provides over the ZOH is negligible.

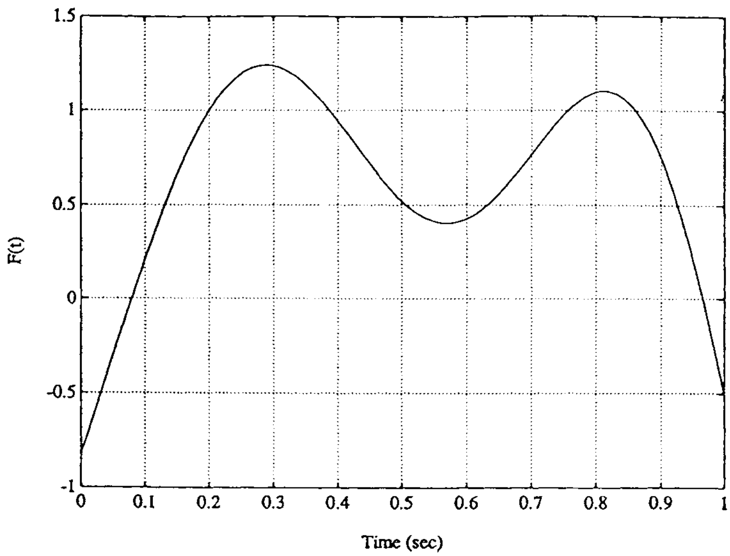


Figure 4 Generalized sampled-data hold function for flexible system example

With these two regulation examples, we have shown that optimizing the hold function may yield substantial gain, as measured by the performance index, and that, sometimes, optimizing the hold function may not yield much improvement, even though the hold function is far from a ZOH. No criteria are known for predicting when optimizing the hold function will yield a significant improvement in performance over that of a ZOH.

B. TRACKING

In this section, we illustrate the use of the results of Section III for the case of output feedback (i.e. $A_C = []$, $B_C = []$, $C_C = []$, and $D_C = K$). First, we consider step function tracking for a second order system. We show that the monodromy boundary condition G_r can be used to affect steady-state error, percent overshoot, and intersample ripple. Next, we compare step function tracking using ZOH and GSHF. We show that under the "fixed monodromy constraint" defined in Section IV A, it may *not* be possible to improve tracking performance. However, using the fact that GSHF transforms the output feedback problem into a state feedback problem we show that it may still be possible to improve tracking performance using GSHF.

1. STEP FUNCTION TRACKING FOR A SECOND ORDER SYSTEM [22]

The plant is a double integrator with position measurement of the form

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{95}$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x.$$

The sampling period is $T = 1$, and we take $D_C = 1$.

The reference model is a unit step function as follows

$$\dot{x}_r = 0, \quad x_r(0) = 1, \tag{96}$$
$$y_r = x_r.$$

The open loop monodromy matrix of the plant, $\Phi(T,0)$, has eigenvalues at 1 and 1, which implies that the plant is not asymptotically stable. Using GSHF control, we choose the matrix G to deadbeat the plant, *i.e.* $\Phi(T,0) + GC$ has all its eigenvalues at the origin. Since the monodromy matrix of the reference model, $\Phi_r(T,0)$, has a single eigenvalue at 1 and since the plant has a single input and a single output, Proposition 3.3 implies that the necessary and sufficient condition to suppress the dynamics of the reference model from the error is to choose G_r such that $\Phi(T,0)+GC+G_rC$ has one eigenvalue equal to 1. The other eigenvalue can be chosen arbitrarily. The output time histories for different chosen eigenvalues are shown in Figure 5. The optimal GSHFs which perform the step function tracking were computed using Proposition 3.4 and are shown in Figures 6. This example shows that by properly using the remaining freedom (*i.e.* the closed loop eigenvalue other than 1) it is possible to remove the ripple between sampling instants. In addition, there is a tradeoff between the intersample steady-state error and the percent overshoot. To reduce intersample steady-state error, one must be willing to accept a larger overshoot.

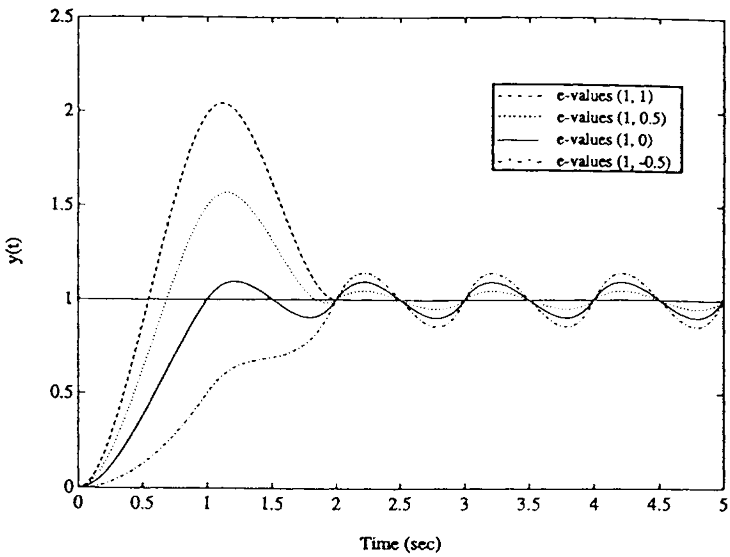


Figure 5 Output time histories of a second order system tracking a step function for different specified eigenvalues

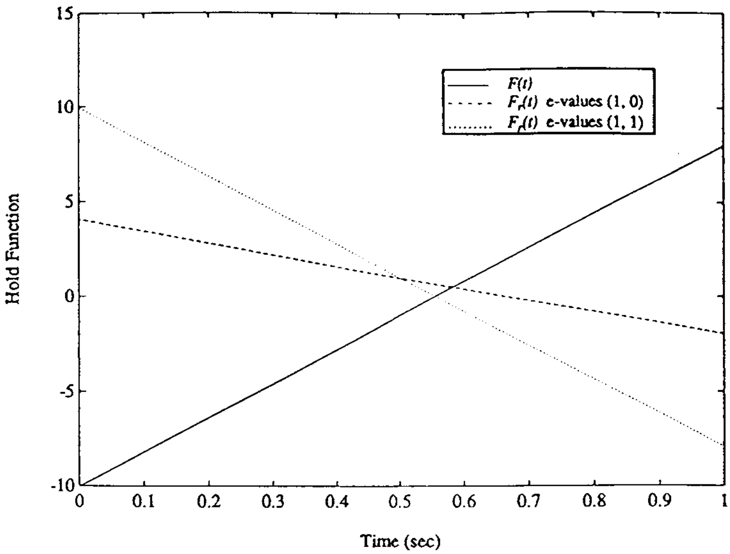


Figure 6 GSHFs for a second order system to track a step function for different specified eigenvalues

2. STEP FUNCTION TRACKING FOR A THIRD ORDER SYSTEM

Consider the plant with an integrator described by

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 4 \\ 0 & -4 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} u, \quad x(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{97}$$

$$y = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} x,$$

with sampling period $T = 1$ and $D_C = -1$. The reference model is once again given by Eq. (96). To achieve tracking at the sampling instants, we consider three hold function configurations: 1) the hold functions are ZOHs, 2) the hold functions are GSHFs under the "fixed monodromy constraint", and 3) the hold functions are GSHFs where G and G_r are design variables subject to Proposition 3.3.

In the first case, the hold functions $F(t)$ and $F_r(t)$ are replaced by ZOHs. Asymptotic tracking at sampling instants is guaranteed by Proposition 3.3 in this case since the matrix $\Phi(T,0)+GC$ is Schur and the matrix $\Phi(T,0)+GC+G_rC$ has one eigenvalue at 1. The output time history is shown in Figure 7.

For the next case, we use GSHF control under the fixed monodromy constraint defined in Section IV A. Since the discretized plant in this case is the same as the discretized plant in the first case, any improvement in tracking performance would be due to better intersampling behavior. However, the output time history for this configuration was identical to the ZOH configuration described above. Moreover, the optimal GSHFs determined from Proposition 3.4 were virtually identical to ZOHs as shown in Figure 8.

For the final case, we use the fact that under GSHF control the output feedback problem is transformed into a state feedback problem to eliminate steady-state ripple in response to the step input. First, we choose G such that the matrix $\Phi(T,0)+GC$ has all its eigenvalues at the origin. Then, we choose G_r such that the matrix $\Phi(T,0)+GC+G_rC$ has the same eigenvalues as $\Phi(T,0)$. Since $\Phi(T,0)$ has an eigenvalue at 1 and the eigenvalue of $\Phi_r(T,0)$ is at 1, this choice for G_r satisfies Proposition 3.3. The hold functions $F(t)$ and $F_r(t)$ are then calculated using Proposition 3.4. Under these conditions, the intersample ripple at steady-state is zero. The output time history is shown in Figure 7 and the corresponding hold functions are shown in Figure 8.

In the first regulation example, we showed that in certain cases an advantage of GSHF control is better intersampling behavior. This example shows that another potential advantage of GSHF control is the ability to choose the monodromy boundary condition.

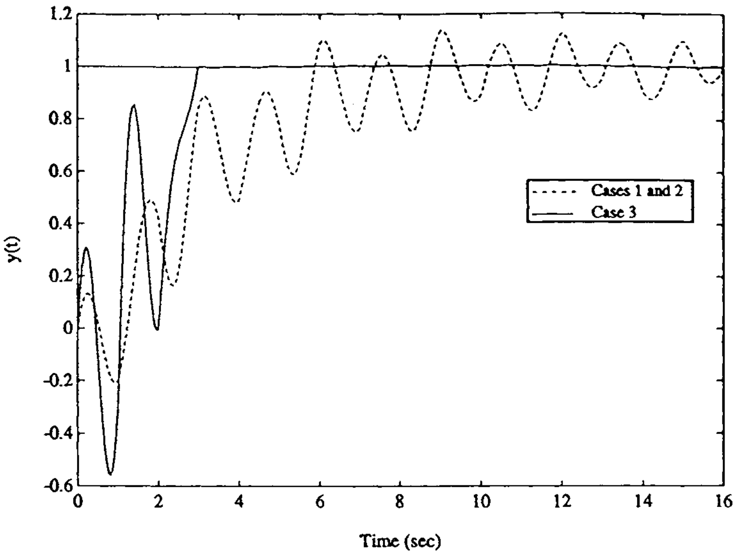


Figure 7 Output time histories of a third order system tracking a step function for different monodromy boundary conditions

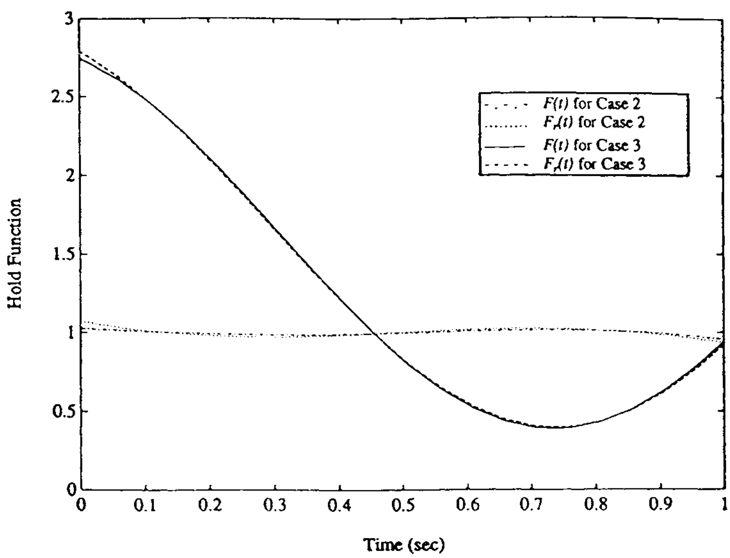


Figure 8 GSHFs for a third order system to track a step function for different monodromy boundary conditions

V. CONCLUSIONS

In this paper, the solutions to two classes of optimal design problems in GSHF control have been presented: regulation and tracking in a standard 4-block configuration. The basic premise is that the hold function itself is the design variable. The distinguishing feature of the design procedure is that the performance index explicitly penalizes the intersampling behavior of the closed loop system by using penalty indices that are time integrals of a quadratic function of the state vector and control input of the analog plant.

For regulation, we have presented solutions to the optimal LQG and LQ regulation problems. It was found that for fixed monodromy problems, we can not only guarantee existence and uniqueness of an optimal hold function, but we can compute it directly by solving a linear two point boundary value problem of Hamiltonian structure. We also have provided a sufficient condition under which the optimal GSHF is exactly a ZOH. Several examples were presented to illustrate the point that, under certain conditions, regulation by optimal GSHF control can result in significant improvement of performance, as measured by the performance index; however, in other cases the optimal GSHF is far from a ZOH with corresponding little improvement in performance. A limitation of this result is that it is not known *a priori* the circumstances under which the optimal GSHF will give a significant improvement in performance.

We have also presented a solution to the deterministic tracking problem using GSHF control. In this section, a necessary and sufficient condition was given for suppressing the dynamics of the reference model from the error signal at the sample times. In addition, an optimal hold function for minimizing a criterion that reflects the intersample output tracking error and the control energy is proved to exist and be unique. A limitation of this result is that the output of the system only asymptotically tracks the reference input at the sampling times. Because of the open loop nature of the control between samples, this is the best one can hope to accomplish with GSHF control. The examples suggest that if the "degree of indeterminacy" defined in Remark 3.1 is greater than one, then we have additional freedom in choosing the matrix G_r. Based on the examples, we expect that this additional freedom can be used to improve measures of system performance other than discrete time tracking.

VI. REFERENCES

1. G. Franklin, J. Powell, and M. Workman, *Digital Control of Dynamic Systems*, Addison-Wesley, Reading, MA, (1990).

2. A. Chammas and C. Leondes, "On the Design of Linear Time Invariant Systems by Periodic Output Feedback, Parts I and II", *International Journal of Control* **27**, pp. 885-903 (1978).

3. A. Chammas and C. Leondes, "Optimal Control of Stochastic Linear Systems by Discrete Output Feedback", *IEEE Transactions on Automatic Control* **AC-23**, pp. 921-926 (1978).

4. A. Chammas and C. Leondes, "On the Finite Time Control of Linear Systems by Piecewise Constant Output Feedback", *International Journal of Control* **30**, pp. 227-234 (1979).

5. P. Kabamba, "Monodromy Eigenvalue Assignment in Linear Periodic Systems", *IEEE Transactions on Automatic Control* **AC-31**, pp. 950-952 (1986).

6. Y. Juan and P. Kabamba, "Constrained Structure Control of Periodic Systems Using Generalized Sampled-Data Hold Functions", *Proceedings of the 1987 American Control Conference*, Minneapolis, MN, (1987).

7. P. Kabamba, "Control of Linear Systems Using Generalized Sampled-Data Hold Functions", *IEEE Transactions on Automatic Control* **AC-32**, pp. 772-783 (1987).

8. T. Kaczorek, "Pole Assignment for Linear Discrete-Time Systems by Periodic Output Feedbacks", *System Control Letters* **6**, pp. 267-269 (1985).

9. T. Kaczorek, "Deadbeat Control of Linear Discrete-Time Systems by Periodic Output-Feedback", *IEEE Transactions on Automatic Control* **AC-31**, pp. 1153-1156 (1986).

10. J. Zavgren, *Stabilisation of Infinite-Dimensional Systems Via Periodic Output Feedback*, Sc.D. Dissertation, Washington University, Saint Louis, MO, (1983).

11. X. Zang, *Sampled Output Stabilization of Infinite-Dimensional Systems*, Sc.D. Dissertation, Washington University, Saint Louis, MO, (1985).

12. J. Zavgren and T. Tarn, "Periodic Output Feedback Stabilization of Infinite-Dimensional Systems", *Proceedings of the 9th IFAC Triennial World Congress*, Budapest, Hungary, (1984).

13. T. Tarn, J. Zavgren, and X. Zeng, "Stabilization of Infinite Dimensional Systems With Periodic Feedback Gains and Sampled Output", *Automatica* **24**, pp. 95-99 (1988).

14. C. Yang, *Control of Linear Systems with Plant Uncertainties Using Generalized Sampled-Data Control Techniques*, Ph.D. Dissertation, University of Michigan, (1990).

15. A. Feuer and G. Goodwin, *Frequency Domain Analysis of Robustness, Sensitivity and Intersample Difficulties*, Technical Report, Department of Electrical and Computer Engineering, University of Newcastle, Australia, (1992).

16. W. Yan and R. Bitmead, "On Applications of Generalized Sampled-Data Hold Functions to Robustness Enhancement", *Proc. 12th IFAC Congress*, Sydney, Australia, (1993).

17. J. Zhang and C. Zhang, "Robustness Analysis of Control Systems Using Generalized Sample Hold Functions", *Proc. 33rd IEEE Conf. Dec. Contr.*, Lake Buena Vista, FL, (1994).

18. P. Dorato and A. Levis, "Optimal Linear Regulators: The Discrete-Time Case", *IEEE Transactions on Automatic Control* **AC-16**, pp. 613-620 (1971).

19. Y. Juan and P. Kabamba, "Optimal Hold Functions for Sampled Data Regulation", *Automatica* **27**, pp. 177-181 (1991).

20. C. Chen, *Linear System Theory and Design*, Harcourt Brace Jovanovich College Publishers, New York, (1984).

21. D. Wilson, "Optimal Solution of Model Reduction Problems", *Proc. IEEE* **17**, pp. 1161-1165 (1970).

22. Y. Juan, *Control of Linear Systems Using Generalized Sampled-Data Hold Functions*, Ph.D. Dissertation, The University of Michigan, (1988).

23. J. Meditch, *Stochastic Optimal Linear Estimation and Control*, McGraw-Hill, , (1969).

24. A. Graham, *Kronecker Products and Matrix Calculus with Applications*, John Wiley and Sons, New York, NY, (1981).

VII. APPENDIX

A. PROOF OF PROPOSITION 2.1

To proceed with the proof of Proposition 2.1, we first need the following preliminary result:

Define

$$L(k) := E\left[x_a(k)x_a^T(k)\right]. \tag{98}$$

Lemma A.1: If the closed loop system, Eq. (17), is stable, we have

$$i) \quad \lim_{k\to\infty} L(k) = L(\infty) = L, \tag{99}$$

$$ii) \quad \lim_{k\to\infty} \frac{1}{k+1}\sum_{i=0}^{k} L(i) = L.$$

Proof of Lemma A.1: Equation (17) represents a stable, linear, time-invariant, discrete-time system driven by white noise. Therefore, *L(k)* satisfies the following Lyapunov equation:

$$L(k+1) = \Psi_a L(k)\Psi_a^T + D_a R_v D_a^T + \begin{bmatrix} R_\omega(T) & 0 \\ 0 & 0 \end{bmatrix}. \tag{100}$$

Since the closed loop system is stable, we know that *L(k)* will converge with exponential rate. As a result, *i)* and *ii)* follow.

∎

Proof of Proposition 2.1: In Eq. (28), the performance index was defined as

$$J = \lim_{t_f \to \infty} E\left\{\frac{1}{t_f}\int_0^{t_f} z^T(t)z(t)dt\right\}. \tag{101}$$

Let $t_f := (k+1)T$ and rewrite Eq. (101) as a sum of integrals as follows:

$$J = \lim_{k \to \infty}\left(\frac{1}{(k+1)T}\sum_{i=0}^{k} J(i)\right), \tag{102}$$

where

$$J(i) := E\left\{\int_{iT}^{(i+1)T} z^T(t)z(t)dt\right\}. \tag{103}$$

From Eqs. (8) and (9), the state and control between sampling instants $t \in [kT,(k+1)T)$ are described by

$$x(t) = \begin{bmatrix}\Phi(t-kT) & D(t-kT)C_C\end{bmatrix}x_a(k) + D(t-kT)D_C v(k) + \omega(t), \tag{104}$$

$$u(t) = \begin{bmatrix}F(t)D_C C & F(t)C_C\end{bmatrix}x_a(k) + F(t)D_C v(k), \tag{105}$$

where x_a is defined in Eq. (13).

Now, consider Eq. (103) and shift the initial time by iT and expand the quadratic term using Eqs. (6), (7), (104), and (105). The result is as follows:

$$\begin{aligned}J(i) = E\Bigg\{\int_0^T \Bigg[x_a^T(i)\Bigg\{&\begin{bmatrix}\Phi^T(t)\\ C_C^T D^T(t)\end{bmatrix}E_1^T E_1\begin{bmatrix}\Phi(t) & D(t)C_C\end{bmatrix} + \\ &\begin{bmatrix}C^T D_C^T\\ C_C^T\end{bmatrix}F^T(t)E_2^T E_2 F(t)\begin{bmatrix}D_C C & C_C\end{bmatrix} + \\ &\begin{bmatrix}\Phi^T(t)\\ C_C^T D^T(t)\end{bmatrix}E_1^T E_2 F(t)\begin{bmatrix}D_C C & C_C\end{bmatrix} + \\ &\begin{bmatrix}C^T D_C^T\\ C_C^T\end{bmatrix}F^T(t)E_2^T E_1\begin{bmatrix}\Phi(t) & D(t)C_C\end{bmatrix}\Bigg\}x_a(i) +\end{aligned}$$

$$\omega^T(t+iT)E_1^T E_1\omega(t+iT)+v^T(i)\Big\{\big(D_C^T D^T(t)E_1^T +$$

$$D_C^T F^T(t)E_2^T\big)\big(E_1 D(t)D_C + E_2 F(t)D_C\big)\Big\}v(i)+$$

$$2x_a^T(i)\left\{\begin{bmatrix}\Phi^T(t)\\ C_C^T D^T(t)\end{bmatrix}E_1^T E_1 D(t)D_C +\right.$$

$$\begin{bmatrix}\Phi^T(t)\\ C_C^T D^T(t)\end{bmatrix}E_1^T E_2 F(t)D_C + \begin{bmatrix}C^T D_C^T\\ C_C^T\end{bmatrix}F^T(t)E_2^T E_1 D(t)D_C +$$

$$\left.\begin{bmatrix}C^T D_C^T\\ C_C^T\end{bmatrix}F^T(t)E_2^T E_2 F(t)D_C\right\}v(i)+2x_a^T(i)\left\{\begin{bmatrix}\Phi^T(t)\\ C_C^T D^T(t)\end{bmatrix}E_1^T E_1 +\right.$$

$$\left.\begin{bmatrix}C^T D_C^T\\ C_C^T\end{bmatrix}F^T(t)E_2^T E_1\right\}\omega(t+iT)+2v^T(i)\Big\{D_C^T D^T(t)E_1^T E_1 +$$

$$D_C^T F^T(t)E_2^T E_1\Big\}\omega(t+iT)\Big]dt\Big\}. \tag{106}$$

Since Eq. (106) is a scalar equation, take the trace and apply the fact that for conformable rectangular matrices A and B $tr(AB)=tr(BA)$. Also, apply the definitions given in Eqs. (98) and (19)-(27). This results in

$$J(i)=tr\big[L(i)M(T)\big]+tr\big[E_1^T E_1 P(T)\big]+tr\big[R_v N(T)\big], \tag{107}$$

where $M(T)$, $P(T)$, and $N(T)$ are defined as follows:

$$\begin{aligned}M(T)=\int_0^T\Bigg\{&\begin{bmatrix}e^{A^T t}+C^T D_C^T D^T(t)\\ C_C^T D^T(t)\end{bmatrix}E_1^T E_1\begin{bmatrix}e^{At}+D(t)D_C C & D(t)C_C\end{bmatrix}+\\ &\begin{bmatrix}e^{A^T t}+C^T D_C^T D^T(t)\\ C_C^T D^T(t)\end{bmatrix}E_1^T E_2\begin{bmatrix}F(t)D_C C & F(t)C_C\end{bmatrix}+\\ &\begin{bmatrix}C^T D_C^T F^T(t)\\ C_C^T F^T(t)\end{bmatrix}E_2^T E_1\begin{bmatrix}e^{At}+D(t)D_C C & D(t)C_C\end{bmatrix}+\\ &\begin{bmatrix}C^T D_C^T F^T(t)\\ C_C^T F^T(t)\end{bmatrix}E_2^T E_2\begin{bmatrix}F(t)D_C C & F(t)C_C\end{bmatrix}\Bigg\}dt,\end{aligned} \tag{108}$$

$$P(T) = \int_0^T R_\omega(t)dt, \tag{109}$$

$$N(T) = \int_0^T \Big\{ D_C^T D^T(t) E_1^T E_1 D(t) D_C + D_C^T F^T(t) E_2^T E_2 F(t) D_C + \\ D_C^T F^T(t) E_2^T E_1 D(t) D_C + D_C^T D^T(t) E_1^T E_2 F(t) D_C \Big\} dt. \tag{110}$$

Finally, from Eqs. (102), (107), and (99), we obtain the following result:

$$J = \lim_{k\to\infty}\left(\frac{1}{(k+1)T} \sum_{i=0}^{k} J(i) \right) = \lim_{k\to\infty}\left(\frac{1}{T} J(k) \right). \tag{111}$$

Along with Eq. (103), the above result implies that

$$J = \lim_{k\to\infty} E\left\{ \frac{1}{T} \int_{kT}^{(k+1)T} z^T(t) z(t) dt \right\}. \tag{112}$$

■

B. PROOF OF PROPOSITION 2.2

We get Eq. (30) by substituting Eq. (107) into Eq. (102) and applying Lemma A.1. Equations (31)-(34) are obtained by writing Eq. (10) and Eqs. (108)-(110) in differential form. Equation (35) is a Lyapunov differential equation obtained from Eq. (27), and Eq. (36) is the steady-state version of the Lyapunov equation given in Eq. (100).

■

C. PROOF OF PROPOSITION 2.3

The minimization of Eq. (30) subject to Eqs. (31)-(36) is performed using Lagrange multipliers. The Lagrangian is defined as follows:

$$
\begin{aligned}
H = {} & \frac{1}{T} tr\left\{LM(T) + E_1^T E_1 P(T) + R_v N(T)\right\} + tr\left[\Theta^T(t)\left(AD(t) + BF(t)\right)\right] + \\
& tr\left\{\Lambda^T(t)\left(\begin{bmatrix} e^{A^T t} + C^T D_C^T D^T(t) \\ C_C^T D^T(t) \end{bmatrix} E_1^T E_1 \begin{bmatrix} e^{At} + D(t) D_C C & D(t) C_C \end{bmatrix} + \right.\right. \\
& \begin{bmatrix} e^{A^T t} + C^T D_C^T D^T(t) \\ C_C^T D^T(t) \end{bmatrix} E_1^T E_2 \begin{bmatrix} F(t) D_C C & F(t) C_C \end{bmatrix} + \\
& \begin{bmatrix} C^T D_C^T F^T(t) \\ C_C^T F^T(t) \end{bmatrix} E_2^T E_1 \begin{bmatrix} e^{At} + D(t) D_C C & D(t) C_C \end{bmatrix} + \\
& \left.\left. \begin{bmatrix} C^T D_C^T F^T(t) \\ C_C^T F^T(t) \end{bmatrix} E_2^T E_2 \begin{bmatrix} F(t) D_C C & F(t) C_C \end{bmatrix}\right)\right\} + \\
& tr\left\{\Omega^T(t)\left[D_C^T D^T(t) E_1^T E_1 D(t) D_C + D_C^T F^T(t) E_2^T E_2 F(t) D_C + \right.\right. \\
& \left.\left. D_C^T F^T(t) E_2^T E_1 D(t) D_C + D_C^T D^T(t) E_1^T E_2 F(t) D_C\right]\right\} + \\
& tr\left\{\Sigma^T(t) R_\omega(t)\right\} + tr\left\{\Gamma^T(t)\left[A R_\omega(t) + R_\omega(t) A^T + D_1 R_w D_1^T\right]\right\} + \\
& tr\left\{K^T\left(\begin{bmatrix} e^{AT} + D(T) D_C C & D(T) C_C \\ B_C C & A_C \end{bmatrix} L \begin{bmatrix} e^{A^T T} + C^T D_C^T D^T(T) & C^T B_C^T \\ C_C^T D^T(T) & A_C^T \end{bmatrix} + \right.\right. \\
& \left.\left. \begin{bmatrix} D(T) D_C \\ B_C \end{bmatrix} R_v \begin{bmatrix} D_C^T D^T(T) & B_C^T \end{bmatrix} + \begin{bmatrix} R_\omega(T) & 0 \\ 0 & 0 \end{bmatrix} - L\right)\right\},
\end{aligned}
\tag{113}
$$

where

$\Theta(t) \in \Re^{n \times r}$, $\Lambda(t) \in \Re^{(n+q)\times(n+q)}$, $\Omega(t) \in \Re^{p \times p}$, $\Sigma(t) \in \Re^{n \times n}$, $\Gamma(t) \in \Re^{n \times n}$, and $K \in \Re^{(n+q)\times(n+q)}$ are matrices of Lagrange multipliers. The necessary conditions for optimality are:

$$
\begin{aligned}
\dot{\Theta}(t) = {} & -\frac{\partial H}{\partial D(t)} = -A^T \Theta(t) - 2 E_1^T E_1 \begin{bmatrix} e^{At} + D(t) D_C C & D(t) C_C \end{bmatrix} \times \\
& \Lambda(t) \begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} - 2 E_1^T E_2 F(t) \begin{bmatrix} D_C C & C_C \end{bmatrix} \Lambda(t) \begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} - \\
& 2 E_1^T E_1 D(t) D_C \Omega(t) D_C^T - 2 E_1^T E_2 F(t) D_C \Omega(t) D_C^T,
\end{aligned}
\tag{114}
$$

$$\dot{\Lambda}(t) = -\frac{\partial H}{\partial M(t)} = 0, \tag{115}$$

$$\dot{\Omega}(t) = -\frac{\partial H}{\partial N(t)} = 0, \tag{116}$$

$$\dot{\Sigma}(t) = -\frac{\partial H}{\partial P(t)} = 0, \tag{117}$$

$$\dot{\Gamma}(t) = -\frac{\partial H}{\partial R_\omega(t)} = -\Sigma(t) - A^T\Gamma(t) - \Gamma(t)A, \tag{118}$$

$$\begin{aligned}\frac{\partial H}{\partial L} = M(T) + &\begin{bmatrix} e^{A^T T} + C^T D_C^T D^T(T) & C^T B_C^T \\ C_C^T D^T(T) & A_C^T \end{bmatrix} \times \\ &K \begin{bmatrix} e^{AT} + D(T)D_C C & D(T)C_C \\ B_C C & A_C \end{bmatrix} - K = 0,\end{aligned} \tag{119}$$

$$\begin{aligned}\frac{\partial H}{\partial F(t)} = B^T\Theta(t) + 2E_2^T E_2 F(t)\left[D_C C \quad C_C\right]\Lambda^T(t)\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + \\ 2E_2^T E_1\left[e^{At} + D(t)D_C C \quad D(t)C_C\right]\Lambda(t)\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + \\ 2E_2^T E_2 F(t) D_C \Omega^T(t) D_C^T + 2E_2^T E_1 D(t) D_C \Omega^T(t) D_C^T = 0,\end{aligned} \tag{120}$$

with boundary conditions

$$\begin{aligned}\Theta(T) = -\frac{\partial H}{\partial D(T)} = -2\left[I_n \quad 0\right]K\Bigg\{&\begin{bmatrix} e^{AT} + D(T)D_C C & D(T)C_C \\ B_C C & A_C \end{bmatrix} \times \\ &L\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + \begin{bmatrix} D(T)D_C \\ B_C \end{bmatrix} R_v D_C^T \Bigg\},\end{aligned} \tag{121}$$

$$\Lambda(T) = -\frac{\partial H}{\partial M(T)} = -L, \tag{122}$$

$$\Omega(T) = -\frac{\partial H}{\partial N(T)} = -R_v, \tag{123}$$

$$\Sigma(T) = -\frac{\partial H}{\partial P(T)} = -E_1^T E_1, \tag{124}$$

$$\Gamma(T) = -\frac{\partial H}{\partial R_\omega(T)} = -\begin{bmatrix} I & 0 \end{bmatrix} K \begin{bmatrix} I \\ 0 \end{bmatrix}. \tag{125}$$

From Eq. (115)-(117), we know that $\Lambda(t)$, $\Omega(t)$, and $\Sigma(t)$ are all equal to constants. These constants are determined from the boundary conditions given in Eqs. (122)-(124). The result is shown below

$$\Lambda(t) = -L, \tag{126}$$

$$\Omega(t) = -R_v, \tag{127}$$

$$\Sigma(t) = -E_1^T E_1. \tag{128}$$

Now, substitute Eqs. (126)-(128) into Eqs. (114)-(120), where appropriate, and Eqs. (37)-(40) result.

∎

D. PROOF OF PROPOSITION 2.4

We follow the same procedure as in the proof of Proposition 2.3, but the boundary condition $\Theta(T)$ is free in this case. Equations (42)-(44) are then obtained by algebraic manipulations. For instance, Eq. (44) for the hold function $F(t)$, $t \in [0, T)$ is obtained by solving Eq. (37) for $F(t)$, and Eqs. (42) and (43) are obtained by substituting the resulting expression for $F(t)$ into Eqs. (31) and (38), respectively.

■

E. PROOF OF PROPOSITION 2.5

The equations for the fixed monodromy problem with parameterized plant dynamics and state weighting are as follows:

$$\dot{D}(t) = \varepsilon A D(t) + BF(t); \quad D(0) = 0, \quad D(T) = G, \tag{129}$$

$$\dot{\Theta}(t) = -\varepsilon A^T \Theta(t) + 2\varepsilon^2 E_1^T E_1 \left[e^{\varepsilon At} + D(t) D_C C \quad D(t) C_C \right] L \begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + \tag{130}$$
$$2\varepsilon^2 E_1^T E_1 D(t) D_C R_v D_C^T + 2\varepsilon E_1^T E_2 F(t) Q,$$

$$F(t) = \left(E_2^T E_2 \right)^{-1} \left\{ \frac{1}{2} B^T \Theta(t) - \varepsilon E_2^T E_1 \left(\left[e^{\varepsilon At} + D(t) D_C C \quad D(t) C_C \right] \times \right. \right. \tag{131}$$
$$\left. \left. L \begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + D(t) D_C R_v D_C^T \right) \right\} Q^{-1}.$$

For the limiting case of $\varepsilon = 0$ the fixed monodromy problem reduces to

$$\dot{D}(t) = \frac{1}{2} B \left(E_2^T E_2 \right)^{-1} B^T \Theta(t) Q^{-1}; \quad D(0) = 0, \quad D(T) = G, \tag{132}$$

$$\dot{\Theta} = 0, \tag{133}$$

$$F(t) = \frac{1}{2} \left(E_2^T E_2 \right)^{-1} B^T \Theta(t) Q^{-1}. \tag{134}$$

Solving the above system of equations results in

$$D(t) = \frac{1}{2} B \left(E_2^T E_2 \right)^{-1} B^T \Theta(0) Q^{-1} t; \quad D(T) = G, \tag{135}$$

$$\Theta(t) = \Theta(0), \tag{136}$$

$$F(t) = \frac{1}{2}\left(E_2^T E_2\right)^{-1} B^T \Theta(0) Q^{-1} \tag{137}$$

Since the discretized plant with GSHF was required to be the same as the discretized plant with ZOH, $G = BT$ for $\varepsilon = 0$. Using this result in Eq. (135) along with the assumption that the columns of B are linearly independent gives

$$\left(E_2^T E_2\right)^{-1} B^T \Theta(0) Q^{-1} = 2I. \tag{138}$$

Equations (137) and (138) result in

$$F(t) = I. \tag{139}$$

As a result, for $\varepsilon = 0$ the optimal hold function is a ZOH. Since ε represents a regular perturbation and since the fixed monodromy problem in Eqs. (129)-(131) is continuous in ε, the optimal GSHF tends to a ZOH as $\varepsilon \to 0$.

∎

F. PROOF OF PROPOSITION 2.6

Rewriting the performance index in Eq. (49) as an infinite sum of integrals and using Eqs. (7), (104), (105), (11), (12), (13), (46), and (108) we obtain

$$J = E\left[\sum_{k=0}^{\infty} x_a^T(kT) M(T) x_a(kT)\right]. \tag{140}$$

Since $x_a(kT+T) = \Psi_a x_a(kT)$, Eq. (A43) becomes

$$J = E\left[\sum_{k=0}^{\infty} x_a^T(0) \Psi_a^{k^T} M(T) \Psi_a^k x_a(0)\right]. \tag{141}$$

Now, define

$$K := \sum_{k=0}^{\infty} \Psi_a^{k^T} M(T) \Psi_a^k. \tag{142}$$

We know that K is the positive semidefinite solution of the Lyapunov equation given in Eq. (40) [23]. The necessary condition for the existence of Eq. (141) is that Ψ_a have spectral radius smaller than one. Using Eq. (47), the performance index becomes

$$J = tr\left(KX_0\right). \tag{143}$$

The minimization of Eq. (143) subject to Eqs. (31), (32), and (40) is performed using Lagrange multipliers. The augmented cost is

$$\begin{aligned}
H = {} & trKX_0 + tr\Theta^T(t)\big(AD(t) + BF(t)\big) + \\
& tr\left\{\Lambda^T(t)\left(\begin{bmatrix} e^{A^T t} + C^T D_C^T D^T(t) \\ C_C^T D^T(t) \end{bmatrix} E_1^T E_1 \begin{bmatrix} e^{At} + D(t)D_C C & D(t)C_C \end{bmatrix} + \right.\right. \\
& \begin{bmatrix} e^{A^T t} + C^T D_C^T D^T(t) \\ C_C^T D^T(t) \end{bmatrix} E_1^T E_2 F(t) \begin{bmatrix} D_C C & C_C \end{bmatrix} + \\
& \begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} F^T(t) E_2^T E_1 \begin{bmatrix} e^{At} + D(t)D_C C & D(t)C_C \end{bmatrix} + \\
& \left.\left. \begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} F^T(t) E_2^T E_2 F(t) \begin{bmatrix} D_C C & C_C \end{bmatrix}\right)\right\} + \\
& tr\left\{L^T\left(\begin{bmatrix} e^{A^T T} + C^T D_C^T D^T(T) & C^T B_C^T \\ C_C^T D^T(T) & A_C^T \end{bmatrix} K \begin{bmatrix} e^{AT} + D(T)D_C C & D(T)C_C \\ B_C C & A_C \end{bmatrix} + \right.\right. \\
& \left.\left. M(T) - K\right)\right\},
\end{aligned} \tag{144}$$

where

$\Theta(t) \in \Re^{n \times r}$, $\Lambda(t) \in \Re^{(n+q)\times(n+q)}$, and $L \in \Re^{(n+q)\times(n+q)}$ are matrices of Lagrange multipliers. The necessary conditions for optimality are:

$$\dot{\Theta}(t) = -\frac{\partial H}{\partial D(t)} = -A^T\Theta(t) - 2E_1^T E_1\left[e^{At} + D(t)D_C C \quad D(t)C_C\right]\times \tag{145}$$
$$\Lambda(t)\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} - 2E_1^T E_2 F(t)\left[D_C C \quad C_C\right]\Lambda(t)\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix},$$

$$\dot{\Lambda}(t) = -\frac{\partial H}{\partial M(t)} = 0, \tag{146}$$

$$\frac{\partial H}{\partial K} = X_0 + \begin{bmatrix} e^{AT} + D(T)D_C C & D(T)C_C \\ B_C C & A_C \end{bmatrix} L \times \tag{147}$$
$$\begin{bmatrix} e^{A^T T} + C^T D_C^T D^T(T) & C^T B_C^T \\ C_C^T D^T(T) & A_C^T \end{bmatrix} - L = 0,$$

$$\frac{\partial H}{\partial F(t)} = B^T\Theta(t) + 2E_2^T E_2 F(t)\left[D_C C \quad C_C\right]\Lambda^T(t)\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} + \tag{148}$$
$$2E_2^T E_1\left[e^{At} + D(t)D_C C \quad D(t)C_C\right]\Lambda(t)\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix} = 0,$$

with boundary conditions

$$\Theta(T) = -\frac{\partial H}{\partial D(T)} = -2\left[I_n \quad 0\right]K\left\{\begin{bmatrix} e^{AT} + D(T)D_C C & D(T)C_C \\ B_C C & A_C \end{bmatrix}\times \right. \tag{149}$$
$$\left. L\begin{bmatrix} C^T D_C^T \\ C_C^T \end{bmatrix}\right\},$$

$$\Lambda(T) = -\frac{\partial H}{\partial M(T)} = -L. \tag{150}$$

From Eqs. (146) and (150), $\Lambda(t) = -L$. Now, substitute $\Lambda(t) = -L$ into Eqs. (145) and (148). The necessary conditions are given by Eqs. (36)-(39) where X_0 replaces

$$\begin{bmatrix} R_\omega(T) & 0 \\ 0 & 0 \end{bmatrix},$$

and $R_v = 0$. $D(t)$, $M(t)$, and K are determined from Eqs. (31), (32), and (40).

∎

G. PROOF OF PROPOSITION 2.7

The proof follows from Propositions 2.4 and 2.6.

∎

H. PROOF OF PROPOSITION 3.1

Let $X_r \in C^{\beta\times\beta}$ be the generalized right eigenvector matrix of $\Phi_r(T,0)$, then

$$\Phi_r(T,0)X_r = X_r\Lambda_r, \tag{151}$$

where Λ_r is in Jordan Canonical form. The generalized right eigenvector matrix of $\Psi_a(T)$ of Eq. (61) corresponding to the eigenvalues of $\Phi_r(T,0)$ has the form $\begin{bmatrix} Y_r \\ W_r \\ X_r \end{bmatrix} = \begin{bmatrix} KX_r \\ K_C X_r \\ X_r \end{bmatrix}$, where $K \in C^{n\times\beta}$ and $K_C \in C^{q\times\beta}$ exist because X_r is nonsingular.

Then, we have

$$\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C & G_r C_r \\ B_C C & A_C & 0 \\ 0 & 0 & \Phi_r(T,0) \end{bmatrix} \begin{bmatrix} KX_r \\ K_C X_r \\ X_r \end{bmatrix} = \begin{bmatrix} KX_r \\ K_C X_r \\ X_r \end{bmatrix} \Lambda_r. \tag{152}$$

Carrying out the matrix multiplication in Eq. (152) and noting that $\Phi_r(T,0)X_r = X_r\Lambda_r$ and X_r is nonsingular results in the following:

$$\begin{aligned} (\Phi(T,0)+GD_C C)K + GC_C K_C + G_r C_r - K\Phi_r(T,0) &= 0, \\ B_C CK + A_C K_C - K_C \Phi_r(T,0) &= 0. \end{aligned} \tag{153}$$

Moreover, since we want to suppress the dynamics corresponding to $\Phi_r(T,0)$ from the error signal, we want, according to Eq. (54)

$$CKX_r - C_r X_r = 0. \tag{154}$$

Noting that X_r is nonsingular gives

$$CK - C_r = 0. \tag{155}$$

Equations (153) and (155) are the necessary and sufficient conditions for suppressing the dynamics of $\Phi_r(T,0)$ from the error signal. Using the property of *vec* operator and Kronecker product [24], we obtain Eq. (62).

∎

I. PROOF OF PROPOSITION 3.2

Suppose $\Phi_r(T,0)$ has a simple eigenvalue λ with right eigenvector x. We have

$$\Phi_r(T,0)x = \lambda x. \tag{156}$$

Since λ is not an eigenvalue of $\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}$, the right eigenvector of $\Psi_a(T)$ corresponding to λ can be computed from

$$\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C & G_r C_r \\ B_C C & A_C & 0 \\ 0 & 0 & \Phi_r(T,0) \end{bmatrix} \begin{bmatrix} y \\ z \\ x \end{bmatrix} = \lambda \begin{bmatrix} y \\ z \\ x \end{bmatrix}. \tag{157}$$

From Eq. (157), y is computed as

$$y = -\begin{bmatrix} I_n & 0_{n\times q} \end{bmatrix} \left\{ \begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix} - \lambda I \right\}^{-1} \begin{bmatrix} G_r C_r \\ 0_{q\times\beta} \end{bmatrix} x. \tag{158}$$

To suppress the dynamics of λ from $e(t)$ at the sampling times, it is necessary and sufficient that

$$C\begin{bmatrix} I_n & 0_{n\times q}\end{bmatrix}\left\{\begin{bmatrix} \Phi(T,0)+GD_CC & GC_C \\ B_CC & A_C \end{bmatrix}-\lambda I\right\}^{-1}\begin{bmatrix} G_rC_r \\ 0_{q\times\beta}\end{bmatrix}x+C_rx=0, \tag{159}$$

or

$$\left\{\begin{bmatrix} C & 0_{p\times q}\end{bmatrix}\left\{\begin{bmatrix} \Phi(T,0)+GD_CC & GC_C \\ B_CC & A_C \end{bmatrix}-\lambda I\right\}^{-1}\begin{bmatrix} G_r \\ 0_{q\times p}\end{bmatrix}+I_p\right\}C_rx=0. \tag{160}$$

Since the reference model was assumed to be observable, $C_rx \neq 0$. As a result,

$$\begin{bmatrix} C & 0_{p\times q}\end{bmatrix}\left\{\begin{bmatrix} \Phi(T,0)+GD_CC & GC_C \\ B_CC & A_C \end{bmatrix}-\lambda I\right\}^{-1}\begin{bmatrix} G_r \\ 0_{q\times p}\end{bmatrix}+I_p \quad \text{is singular.}$$

Now, apply the fact that for two matrices $L\in\Re^{n\times m}$ and $M\in\Re^{m\times n}$ $\det(I_n+LM)=\det(I_m+ML)$ and conclude that

$$\left\{\begin{bmatrix} \Phi(T,0)+GD_CC & GC_C \\ B_CC & A_C \end{bmatrix}-\lambda I\right\}^{-1}\begin{bmatrix} G_r \\ 0_{q\times p}\end{bmatrix}\begin{bmatrix} C & 0_{p\times q}\end{bmatrix}+I_{n+q} \quad \text{is singular.}$$

The above can be rewritten as follows

$$\left\{\begin{bmatrix} \Phi(T,0)+GD_CC & GC_C \\ B_CC & A_C \end{bmatrix}-\lambda I\right\}^{-1}\left\{\begin{bmatrix} G_rC & 0_{n\times q} \\ 0_{q\times n} & 0_{q\times q}\end{bmatrix}+\right.$$
$$\left.\begin{bmatrix} \Phi(T,0)+GD_CC & GC_C \\ B_CC & A_C \end{bmatrix}-\lambda I\right\} \quad \text{is singular.}$$

Since λ is not an eigenvalue of $\begin{bmatrix} \Phi(T,0)+GD_CC & GC_C \\ B_CC & A_C \end{bmatrix}$, this gives

$$\begin{bmatrix} \Phi(T,0)+GD_CC+G_rC & GC_C \\ B_CC & A_C \end{bmatrix}-\lambda I \quad \text{is singular.}$$

Consequently, λ is an eigenvalue of $\begin{bmatrix} \Phi(T,0)+GD_C C+G_r C & GC_C \\ B_C C & A_C \end{bmatrix}$.

∎

J. PROOF OF PROPOSITION 3.3

Suppose $\Phi_r(T,0)$ has a simple eigenvalue λ with right eigenvector x. The necessary and sufficient condition for the output of the system to asymptotically track the reference input at the sampling time is given in Eq. (160).

If λ is an eigenvalue of $\begin{bmatrix} \Phi(T,0)+GD_C C+G_r C & GC_C \\ B_C C & A_C \end{bmatrix}$, then

$\begin{bmatrix} \Phi(T,0)+GD_C C+G_r C & GC_C \\ B_C C & A_C \end{bmatrix}-\lambda I$ is singular.

Since λ is not an eigenvalue of $\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}$,

$\left\{\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}-\lambda I\right\}^{-1}\left\{\begin{bmatrix} \Phi(T,0)+GD_C C+G_r C & GC_C \\ B_C C & A_C \end{bmatrix}-\lambda I\right\}$ is singular,

and

$\left\{\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}-\lambda I\right\}^{-1}\begin{bmatrix} G_r \\ 0_{q\times p} \end{bmatrix}\begin{bmatrix} C & 0_{p\times q} \end{bmatrix}+I_{n+q}$ is singular.

If we are tracking only one signal (i.e. p=1), then

$$\begin{bmatrix} C & 0_{p\times q} \end{bmatrix}\left\{\begin{bmatrix} \Phi(T,0)+GD_C C & GC_C \\ B_C C & A_C \end{bmatrix}-\lambda I\right\}^{-1}\begin{bmatrix} G_r \\ 0_{q\times p} \end{bmatrix}+1=0, \qquad (161)$$

and Eq. (160) is satisfied.

∎

Actuator Placement Using Degree of Controllability for Discrete-Time Systems

Guang-Qian Xing
Department of Mechanical Engineering
Howard University
Washington, DC 20059, USA
and
Beijing Institute of Control Engineering
Beijing, Peoples' Republic of China

Peter M. Bainum
Department of Mechanical Engineering
Howard University
Washington, D.C. 20059, USA

I. INTRODUCTION

The concept of controllability is one of the cornerstones of modern control theory. Yet despite its fundamental importance from a theoretical point of view, its practical utility for control system evaluation is limited due to its binary nature. That is, a system is either controllable or it is uncontrollable. There is no provision for consideration of the more subtle question: how controllable is the system?

The desirability of a degree of controllability (earlier it was called the controllability index) concept has been recognized in the literature since 1961 when Kalman, Ho and Narendra [1] discussed it. Early papers in the area [2-4] concentrated on particular properties of either the controllability Grammian matrix or the controllability matrix itself in developing definitions of the degree of controllability. It is natural to try to connect the degree of controllability to properties of the standard controllability matrix $P_c = (B:AB : ... : A^{n-1}B)$, and define the degree of controllability as the square root of the minimum eigenvalue of $P_cP_c^T$. In 1979 Viswanathan, Longman and Likins considered the "shortcomings" of this definition [5]. Four apparent difficulties with this definition must somehow be handled before the definition becomes "viable"; therefore, they presented a new definition of the degree of controllability of a control system by means of a scalar measure based on the concept of the "recovery region".

In this paper three candidate definitions of degree of controllability, which were presented first by Muller and Weber [4] for linear continuous systems, are presented for linear discrete-time systems based on the scalar measure of the Grammian matrix. The three candidates for the degree of observability of linear discrete-time systems are also presented. Because some difficulties with this definition (as pointed out by Viswanathan [5]) have been handled here, and the degree of controllability (observability) based on a scalar measure of the Grammian matrix can be readily calculated, the three candidates for the degree of controllability (observability) are viable for practical engineering design.

The emphasis of this paper is in showing the physical and geometrical meanings and the general properties of the three candidate definitions of degree of controllability (observability) as physically meaningful measures. The concepts of degree of controllability are applied to the actuator placement problem for the orbiting shallow spherical shell control system. The degree of controllability for seven cases in which the placement of actuators are all different are compared. The LQG transient responses for several typical systems with different controllability are also shown.

II. CONCEPT AND PHYSICAL MEANING OF CONTROLLABILITY

In order to show the physical meaning of the degree of controllability, first of all, we consider the fuel optimal problem for the linear discrete-time system. The state equations are as follows:

$$x(k_i) = \Phi(k_i, k_{i-1})\, x(k_{i-1}) + \Gamma(k_{i-1})\, u(k_{i-1}) \qquad (i = 1, 2, \ldots N) \tag{1}$$

the fuel optimal control problem can be stated as follows: To find the system control $u(k_{i-1})$ $(i = 1, 2, \ldots N)$ which can transfer from the initial state $x(k_0)$ to the final state of the system $x(k_N) = 0$ during the time interval $[k_0, k_N]$, and such that the performance index,

$$J_N = \sum_{i=1}^{N} < u(k_{i-1}),\ u(k_{i-1}) > \tag{2}$$

is minimized.

Equation (1) may be written

$$x(k_N) = \Phi(k_N, k_0)\, x(k_0) + \sum_{i=1}^{N} \Phi(k_N, k_i)\, \Gamma(k_{i-1})\, u(k_{i-1}) \tag{3}$$

or

$$x(k_N) - \Phi(k_N, k_0)\ x(k_0) = T_p U \tag{4}$$

where

$$T_p = (\Gamma(k_{k-1}) \vdots \Phi(k_N, k_{N-1})\, \Gamma(k_{N-2}) \vdots \ldots \ldots \Phi(k_N, k_1)\, \Gamma(k_0)) \tag{5}$$

$$U = \begin{bmatrix} u(k_{N-1}) \\ u(k_{N-2}) \\ \bullet \\ \bullet \\ \bullet \\ u(k_0) \end{bmatrix} \tag{6}$$

Considering the requirement that $x(k_N)=0$, Equation (4) may be written:

$$-x(k_0) = P_c U \tag{7}$$

where

$$P_c = \Phi(k_0, k_N)\, T_p \tag{8}$$

and the fuel optimal problem becomes the following conditional extremum problem

$$inf\ \{\|U\|^2 | P_c U(\bullet) = -x(k_0)\} \tag{9}$$

where $\|\bullet\|$ represents the Euclidian norm. This problem can be solved by using the Lagrange multiplier method. If λ is the Lagrange multiplier, the problem stated in Eq. (9) is equivalent to the general problem of extremizing the following cost function

$$J_p = < U, U > = \lambda^T (P_c U + x(k_0))$$

In order for J_p to be an extremum the following necessary conditions must be satisfied

$$\frac{\partial J_p}{\partial U} = 2U - P_c^T\lambda = 0$$
$$\frac{\partial J_p}{\partial \lambda} = P_c U + x(k_0) = 0 \tag{10}$$

i.e.,

$$(P_c P_c^T)\lambda = -2x(k_0)$$
$$2U = P_c^T\lambda \tag{11}$$

If the system is controllable, then the inverse of the matrix, $(P_cP_c^T)$, exists, and the optimal control, U^*, is

$$U^* = -P_c^T(P_cP_c^T)^{-1}x(k_0) \tag{12}$$

After substituting Eq. (12) into the performance index, Eq. (2), the minimum control energy is obtained as follows:

$$\begin{aligned} J_N^* &= \langle U^*, U^* \rangle \\ &= \langle -P_c^T(P_cP_c^T)^{-1}x(k_0), -P_c^T(P_cP_c^T)^{-1}x(k_0) \rangle \\ &= \langle (P_cP_c^T)^{-1}x(k_0), x(k_0) \rangle \end{aligned} \tag{13}$$

i.e.,

$$J_N^* = x^T(k_0)\, W_c^{-1}(k_N, k_0)\, x(k_0) \tag{14}$$

where, $W_c(k_N,k_0) = P_c P_c^T$

From intuitive considerations, the linear system "1" is more controllable than the linear system "2" if the energy, J^*_{N1}, required by system "1" for transferring the initial state, $X(k_0)$, to the final state $X(k_N) = 0$, is less than the energy required by the second system, J^*_{N2}, for transfer between the same initial and final states within the same time period, or stated mathematically,

$$x^T(k_0)\, W_{C1}^{-1}(k_N, k_0)\, x(k_0) \le x^T(k_0)\, W_{C2}^{-1}(k_N, k_0)\, x(k_0) \qquad (15)$$

$$\forall\ x(k_0) \in R^n$$

Because $W_{c1}^{-1}(k_N, k_0)$ and $W_{c2}^{-1}(k_N, k_0)$ are two symmetrical positive definite matrices, the relationship (15) should be satisfied for any initial condition, $x(k_0) \epsilon R^n$; this implies that

$$W_{C1}^{-1}(k_N, k_0) \le W_{C2}^{-1}(k_N, k_0) \qquad (16)$$

or

$$W_{C1}(k_N, k_0) \ge W_{C2}(k_N, k_0) \qquad (17)$$

Considering the symmetry of the matrices, W_{c1} and W_{c2}, the following relationships should be satisfied:

$$\lambda_i(W_{C1}^{-1}) \le \lambda_i(W_{C2}^{-1}) \qquad (i = 1, \ldots n) \qquad (18)$$

or

$$\lambda_i(W_{C1}) \ge \lambda_i(W_{C2}) \qquad (1 = 1, \ldots n) \qquad (19)$$

where $\lambda_i(W_{C1}^{-1})$, $\lambda_i(W_{C2}^{-1})$, $\lambda_i(W_{C1})$, $\lambda_i(W_{C2})$ are the i^{th} eigenvalues of the matrices, W_{c1}^{-1}, W_{c2}^{-1}, W_{c1} and W_{c2}, respectively. It is evident from Eq. (14) that the loci of equi-control effort (J^*_N = constant) is a super ellipsoid and its equation (J^*_N =1) in the principal axis system is as follows:

$$\frac{x_1^2}{\frac{1}{\lambda_1(W_c^{-1})}} + \frac{x_2^2}{\frac{1}{\lambda_2(W_c^{-1})}} + \ldots + \frac{x_n^2}{\frac{1}{\lambda_n(W_c^{-1})}} = 1 \tag{20}$$

Considering the relationship between $\lambda_i(W_c^{-1})$ and $\lambda_i(W_c)$: :

$$\lambda_i(W_c^{-1}) = \frac{1}{\lambda_{n+1-i}\ (W_c)} \tag{21}$$

i.e.,

$$\lambda_1(W_c^{-1}) < \lambda_2(W_c^{-1}) < \ldots < \lambda_n(W_c^{-1})$$

$$\lambda_1(W_c) = \frac{1}{\lambda_n(W_c^{-1})} < \lambda_2(W_c) = \frac{1}{\lambda_{n-1}(W_c^{-1})} < \ldots$$

$$\lambda_n(W_c) = \frac{1}{\lambda_1(W_c^{-1})}$$

Then Eq. (20) can be written as follows:

$$\frac{x_1^2}{\lambda_1(W_c)} + \frac{x_2^2}{\lambda_2(W_c)} + \ldots + \frac{x_n^2}{\lambda_n(W_c)} = 1 \tag{22}$$

Therefore, from the geometrical viewpoint, it is clear from Eqs. (15) and (22) that if system "1" is more controllable than system "2", each axis of the super ellipsoid for system "1" is longer than the corresponding axis for the system "2". The square root of each of the eigenvalues of the matrix, W_c, is just the length of the corresponding super ellipsoid principal axis.

III. THREE CANDIDATES FOR THE DEFINITIONS OF DEGREE OF CONTROLLABILITY AND THEIR GENERAL PROPERTIES

As we see in Eq. (15), in terms of the matrices this comparison of the costs of the two systems leads to (17), i.e.,

$$W_{c1}(k_N, k_0) \geq W_{c2}(k_N, k_0) \tag{17}$$

which means that $W_{c1}-W_{c2}$ has to be a positive semi-definite matrix. The matrix condition (17) is equivalent to n scalar conditions and, therefore, a simple scalar quantity cannot be employed to describe the conditions of Eq. (17). For engineering purposes it is desirable to replace the matrix measure by a scalar figure of merit[1]. There are three obvious candidates for such scalar measures: the maximum eigenvalue of $W_c^{-1}(k_N, k_0)$, the trace of $W_c^{-1}(k_N, k_0)$ and the determinant of $W_c(k_N, k_0)$[4]. The significance of these quantities is shown in the following section.

A. The first candidate for the degree of controllability, μ_1

As shown in (15), we must consider the values of the control effort $X^T(k_0)W_c^{-1}(k_N, k_0)X(k_0)$, $X(k_0)\epsilon R^n$ when comparing the control effort of the two systems. It is natural to use the maximum control effort,

$$\begin{aligned} &Max \quad x^T(k_o)\, W_c^{-1}(k_N, k_0)\, x(k_0) \\ &\| x(k_0) \| = 1 \\ &x(k_0) \in R^n \end{aligned} \tag{23}$$

as a scalar measure of the control effort. It is well known that the scalar quantity defined in (23) is just the maximum eigenvalue, $\lambda_{max}(W_c^{-1})$, of the matrix, W_c^{-1}, and the $\lambda_{max}(W_c^{-1})$ is just the reciprocal of the minimum eigenvalue of the matrix W_c, i.e.,

$$\begin{aligned} &\underset{\substack{\| x(k_0) \| = 1 \\ x(k_0) \in R^n}}{Max} \; x^T(k_0) W_c^{-1}(k_N, k_0) = \lambda_{\max}(W_c^{-1}) = \frac{1}{\lambda_{\min}(W_c)} \end{aligned} \tag{24}$$

where the $\lambda_{\min}(W_c)$ is the minimum eigenvalue of the matrix, $W_c(k_N, k_0)$. For the reasons given above, we may define the degree of controllability, μ_1, as follows:

$$\mu_1 = \lambda_{\min}(W_c(k_N, k_0)) = \sigma_{\min}^2 (W_c) \tag{25}$$

This definition means that as μ_1 increases, then the control effort decreases. It is evident that the square root of the degree of controllability defined by (25) is just the minimum semi-axis of the super-ellipsoid defined by Eq. (22) (it can be called the degree of controllability super-ellipsoid). If $\mu_1 = 0$, the system will not be controllable.

B. The second candidate for the degree of controllability, μ_2

Similar to (23), we can also use the average value of the control effort, $x^T(k_0)Wc^{-1}x(k_0)$, over the unit hypersphere $\{ x(k_0) : \| x(k_0) \| = 1\}$

$$\frac{\int_{\| x(k_0) \| = 1} x^T(k_0) W_c^{-1}(k_o)\, dx(k_0)}{\int_{\| x(k_0) \| = 1} dx(k_0)}$$

as a scalar measure of the control effort. After integrating, it can be obtained [4]

$$\frac{\int_{\| x(k_0) \| = 1} x^T(k_o) W_c^{-1} x(k_o)\, dx(k_0)}{\int_{\| x(k_0) \| = 1} dx(k_0)} = \frac{1}{n} tr W_c^{-1} \tag{26}$$

where n is the order of the state. Therefore, we may define the second candidate for the degree of controllability as follows:

$$\mu_2 = \frac{n}{trW_c^{-1}(k_N, k_0)} \tag{27}$$

For practical applications it is desirable to maintain the average cost (26) as small as possible. Hence, the measure, μ_2, has to be as large as possible. The definition (27) of a measure instead of (26) is more convenient because uncontrollable systems are characterized by a vanishing value of μ_2, which arises from a limiting process.

C. The third candidate of degree of controllability, μ_3

The third possible candidate for a scalar quantitative measure of controllability is the determinant of $W_c(k_N, k_0)$ because the volume of the degree of the controllability super-ellipsoid, $X^T(k_0)\, W_c^{-1}\,(k_N, k_0)\, x(k_0) = 1$, is proportional to the square root of the determinant of $W_c(k_N, k_0)$:

$$Vol = \int_{x(k_0)^T W_c^{-1} x(k_0) \le 1} dx\,(k_0) = Const\,.\sqrt{detW_c} \tag{28}$$

Therefore, we may define the third candidate definition of degree of controllability as follows:

$$\mu_3 = (detW_c(k_N, k_0))^{1/n} \tag{29}$$

From geometrical considerations, $2^n\,(\mu_3)^{n/2}$ is just the volume of the hyperrectangular parallelepiped whose sides are the axes of the hyperellipsoid. The larger, μ_3, the more controllable the system is; the uncontrollable system is also characterized by $\mu_3 = 0$.

D. The general properties of the degree of controllability

In the previous sections we gave the definitions of degree of controllability and their physical and geometry meaning. The degrees of controllability, as a scalar measure, are invariant under an orthogonal similarity transformation. The proof will be given in the Appendix of this paper. In addition, they are scalar quantities defined over the cone set, W_c^*, of positive definite and semi-definite matrices. Their properties are as follows: [4]

a. $\mu_i(W_c) = 0$ (i = 1, 2, 3) for $W_c \in W_c^*$ with det $W_c = 0$, i.e., the system cannot be controllable;

b. $\mu_i(W_c) > 0$ (i = 1, 2, ... N) for $W_c \in W_c^*$ with det $W_c > 0$;

c. $\mu_i(\lambda W_c) = \lambda \mu_i(W_c) = \lambda \mu_i(W_c)$ (i = 1, 2, 3)

for $\lambda > 0$, $W_c \in W_c^*$;

d. $\mu_i(W_c) \geq \mu_i(W_{c1}) + \mu_i(W_{c2})$ (i = 1, 2, 3)

for $W_c = W_{c1} + W_{c2}$, W_{c1}, $W_{c2} \in W_c^*$.

The proof of the properties (a) - (d) will be given in the Appendix of this paper.

The physical meaning of the properties (a) - (c) are evident. The physical meaning of the property (d) is that the degree of controllability for the system which contains two control elements is higher than the sum of the degrees of controllability for the two systems each of which contain only a single controller.

The μ_1, μ_2, μ_3 are three kinds of candidates for the degree of controllability, the physical meanings of which are different. The following relationship among μ_1, μ_2 and μ_3 can be stated [4]:

$$\mu_1 \leq \mu_2 \leq \mu_3 \tag{29}$$

The sufficient and necessary condition for the equality to be true is that all of the eigenvalues of W_c be equal.

E. The geometrical meaning of the concept of DOC

As we have shown from (22), the loci of equi-control effort($J_N^*=c_1=1$) is a superellipsoid and its equation($J^{N*}=1$) in the principal axis system is

$$\frac{x_1^2}{\lambda_1(W_c)} + \frac{x_2^2}{\lambda_2(W_c)} + \ldots + \frac{x_n^2}{\lambda_n(W_c)} = 1 \tag{22}$$

The control efforts for the points on the superellipsoid are equal, and the control effort for the points inside the superellipsoid are less than the points on the surface of the superellipsoid.

Therefore, the more controllable the control system is, the bigger the superellipsoid of the control effort is. The superellipsoid of the control effort will be called the Degree of Controllability Superellipsoid(DOCS). The matrix $W_c(K_N,K_0)$ can be called the Degree of Controllability Matrix(DOCM). The square root of an eigenvalue of DOCM is just the length of the corresponding principal axes of the DOCS.

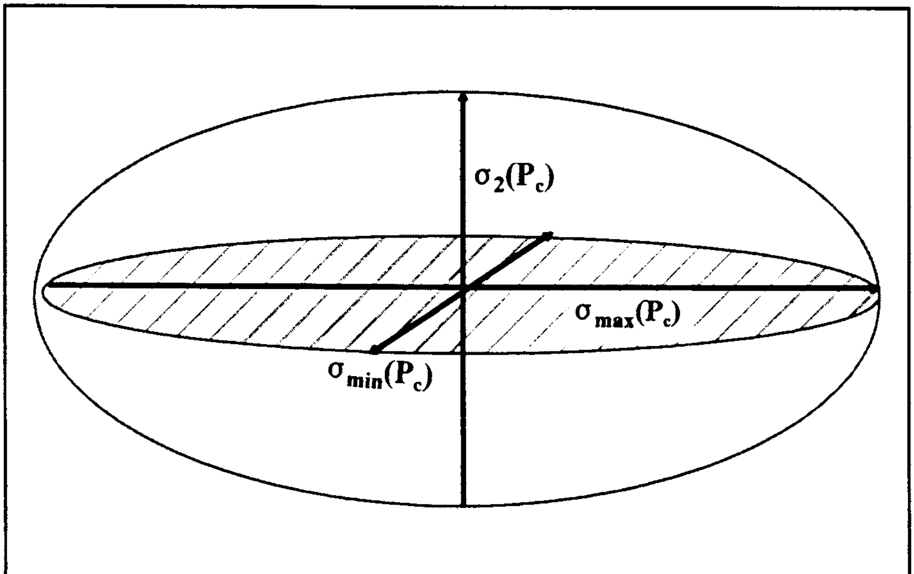
Example: The geometry of DOC superellipsoid for 3-dimension

The equi-control effort equation is

$$X(k_0)^T W_{c-1} X(k_0) = 1$$

In the principal axis system, the equation is

$$\frac{X_1^2(k_0)}{\sigma_{min}^2(P_c)} + \frac{X_2^2(k_0)}{\sigma_2^2(P_c)} + \frac{X_3^2(k_0)}{\sigma_{max}^2(P_c)} = 1$$



where

$\sigma_i(P_c)$: Singular value of the controllable matrix P_c

$P_c=(\Phi(k_0,k_N)\Gamma(k_{N-1}),\ \Phi(k_0,k_{N-1})\Gamma(k_{N-2}),\ \ldots,\ \Phi(k_0,k_1)\Gamma(k_0))$, $W_c=P_cP_c^T$

The geometrical meaning of the definitions of DOC, μ_1, μ_2, μ_3, is shown in theTable 1.

Table 1: The Geometrical Meaning of μ_1, μ_2, μ_3

Definition	Geometrical Meaning
$\mu_1=\lambda_{min}(W_c)=\sigma_{min}^2(P_c)$	the minimum semi-axis of DOCS $\sigma_{min}(P_c)$
$\mu_2=3/trW^{-1}$ $=3/(\sigma_{min}^{-2}+\sigma_2^{-2}+\sigma_{max}^{-2})$	the algebraic average of the semi-axes of DOCS $(\sigma_{min}+\sigma_2+\sigma_{max})/3$
$\mu_3=(\lambda_{min}^2.\lambda_\lambda^2.\lambda_{max})^{1/3}$ $=(\sigma_{min}^2.\sigma_2^2.\sigma_{max}^2)^{1/3}$	the geometrical average of the semi-axes of DOCS $(\sigma_{min}.\sigma_2.\sigma_{max})^{1/3}$

In general, the values of μ_1,μ_2,μ_3 for the two different systems are not consistent. In order to select one system from the two systems, we can use the average of μ_1, μ_2, μ_3 as the criterion for selecting the systems. Sometimes, the selection of μ_1,μ_2,μ_3, as the numerical measure of the system DOC, also depends on the information for the initial state of the system.

IV. THE DEFINITION OF DEGREE OF OBSERVABILITY AND ITS PHYSICAL MEANING

Suppose the dynamical equations and observational equations are as follows:

$$x(k_i) = \Phi(k_i, k_{i-1})\, x(k_{i-1}) + \Gamma(k_{i-1})\, u(k_{i-1}) \tag{30}$$

$$\begin{gathered} y(k_{i-1}) = H(k_{i-1})\, x(k_{i-1}) + v(k_{i-1}) \\ (i = 1, 2, \ldots N) \end{gathered} \tag{31}$$

where $v(k_{i-1})$ is observational white noise with statistical properties

$$Ev\{(k_{i-1})\} = 0; \;\; E\{v(k_{i-1})\, v^T(k_{i-1})\} = R_{i-1}$$

The equation (30) may be written in another form in terms of the initial state as follows.

$$x(k_i) = \Phi(k_i, k_0)\, x(k_0) + \sum_{j=1}^{i} \Phi(k_i, k_j)\, \Gamma(k_{j-1})\, u(k_{j-1}) \tag{32}$$

$$\begin{aligned} y(k_{i-1}) &= H(k_{i-1})\, \Phi(k_{i-1}, k_0)\, x(k_0) + \\ &\quad H(k_{i-1}) \sum_{j=1}^{i-1} \Phi(k_{i-1}, k_j)\, \Gamma(k_{j-1})\, u(k_{j-1}) + v(k_{j-1}) \\ &\qquad (i = 1, 2, \ldots N) \end{aligned} \tag{33}$$

When i assumes values from 1 through N, Eq. (33) can be put in a matrix form as follows:

$$Y = P_o x(k_0) + V \tag{34}$$

where

$$Y = \begin{bmatrix} y(k_0) \\ y(k_1) - H(k_1)\Gamma(k_0)u(k_0) \\ \vdots \\ y(k_{N-1}) - H(k_{N-1}) \sum_{j=1}^{N-1} \Phi(k_{N-1}, k_j)\Gamma_{j-1}u(k_{j-1}) \end{bmatrix}$$

$$P_o = \begin{bmatrix} H(k_0) \\ H(k_1)\Phi(k_1, k_0) \\ \vdots \\ H(k_{N-1})\Phi(k_{N-1}, k_0) \end{bmatrix} \qquad V = \begin{bmatrix} v(k_0) \\ v(k_1) \\ v(k_2) \\ \vdots \\ v(k_{N-1}) \end{bmatrix}$$

$$E\{V\} = 0 \qquad E\{VV^T\} = \begin{bmatrix} R_0 & & & \\ & R_1 & & \\ & & \ddots & \\ & & & R_{N-1} \end{bmatrix} = R$$

We consider the weighted least square estimate problem; the performance index of the weighted least square estimate is

$$J = (Y - P_o x(k_0))^T R^{-1} (Y - P_o x(k_0)) \tag{35}$$

It is well known that the weighted least square estimate $x(k_0)$ is

$$\hat{x}(k_0) = (P_o^T R^{-1} P_o)^{-1} P_o^T R^{-1} Y \tag{36}$$

The estimate error is

$$\begin{aligned} x(k_0) - \hat{x}(k_0) &= (P_o^T R^{-1} P_o)^{-1} P_o R^{-1} (P_o x(k_o) - Y) \\ &= -(P_o^T R^{-1} P_o)^{-1} P_o^T R^{-1} V \end{aligned}$$

The covariance of the estimate error is

$$E\{(x(k_0)-\hat{x}(k_0))(x(k_0)-\hat{x}(k_0))^T\} = (P_o^T R^{-1} P_o)^{-1} \tag{37}$$

When R = I, the covariance of the least square estimate error is just

$$\begin{aligned} &E\{(x(k_0) - \hat{x}(k_0))(x(k_0)-\hat{x}(k_0))^T\} \\ &= (P_o^T P_o)^{-1} = W_0^{-1} \end{aligned} \tag{38}$$

where

$$W_0 = P_o^T P_o \tag{39}$$

Similar to the degree of controllability, we call the W_0 the degree of the observability matrix. The three kinds of definitions of degree of observability are as follows:

$$\mu_1(W_0) = \lambda_{min}(W_0) \tag{40}$$

$$\mu_2(W_0) = \frac{n}{trW_0^{-1}} \tag{41}$$

$$\mu_3(W_0) = (\det W_0)^{\frac{1}{n}} \tag{42}$$

Because the W_0^{-1} is the covariance matrix of the least square estimate, then the trW_0^{-1} is the sum of the squares of the variances associated with the least square error estimate, so the physical meaning of $\mu_2(W_0)$ is evident. $\mu_1(W_0)$ is the maximum eigenvalue of the covariance error matrix W_0^{-1} for the least square estimate. $\mu_3(W_0)$ is the reciprocal of the geometrical average of the error covariance matrix eigenvalues for the least square estimate.

In general, if the estimation accuracy of system "1" is higher than that of system "2", it means

$$W_{01}^{-1} \le W_{02}^{-1} \tag{43}$$

t implies

$$W_{01} \ge W_{02} \qquad trW_{01}^{-1} \le trW_{02}^{-1} \tag{44}$$

and

$$\lambda_i(W_{01}) \ge \lambda_i(W_{02}) \qquad (i = 1,\ 2,\ \ldots\ n) \tag{45}$$

Therefore

$$\mu_1(W_{01}) \geq \mu_1(W_{02})$$

$$\mu_2(W_{01}) \geq \mu_2(W_{02})$$

$$\mu_3(W_{01}) \geq \mu_3(W_{02})$$

i.e., the degrees of observability defined by (40-42) for system "1" are all higher than the corresponding degree of observability for system "2".

V. The Degree of Controllability (Observability) for Discrete Time-Invariant Systems

Let the dynamical system and observational system for the discrete time-invariant system be described as follows:

$$x(k_i) = \Phi x(k_{i-1}) + \Gamma u(k_{i-1}) \tag{46}$$

$$y(k_i) = Hx(k_i) + v(k_i) \tag{47}$$

The degree of controllability matrix, W_c, for the system (46), (47) is

$$W_c = P_c P_c^T \tag{48}$$

where

$$P_c = (\Gamma \vdots \Phi\Gamma \vdots \;\ldots\; \Phi^{n-1}\Gamma) \tag{49}$$

The degree of observability matrix for system (46), (47) is

$$W_0 = P_o^T P_o \tag{50}$$

where

$$P_0 = \begin{bmatrix} H \\ H\phi \\ \vdots \\ H\phi^{n-1} \end{bmatrix} \tag{51}$$

The three kinds of definitions of degree of controllability are as follows:

$$\mu_1(W_c) = \lambda_{\min}(W_c) \tag{52}$$

$$\mu_2(W_c) = \frac{n}{trW_c^{-1}} \tag{53}$$

$$\mu_3(W_c) = (\det W_c)^{\frac{1}{n}} \tag{54}$$

Similarly, the three kinds of definitions of the degree of observability are as follows:

$$\mu_1(W_0) = \lambda_{\min}(W_0) \tag{55}$$

$$\mu_2(W_0) = \frac{n}{trW_0^{-1}} \tag{56}$$

$$\mu_3(W_0) = (\det W_0)^{\frac{1}{n}} \tag{57}$$

VI. APPLICATION

We would like to use the concepts of the degree of controllability for the actuator placement problem of the shallow spherical shell.

The motion equations for a shallow spherical shell in orbit subject to external forces and torques are as follows [6]:

$$\dot{X} = AX + Bu \tag{58}$$

where

$$A = \begin{bmatrix} 0 & I \\ -\bar{A} & D \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \bar{B} \end{bmatrix}$$

$$X = (\psi,\ \phi,\ \theta, \varepsilon_1,\ \varepsilon_2,\ \ldots,\ \varepsilon_6,\ \psi,\ \dot{\phi},\ \dot{\theta},\ \dot{\varepsilon}_1,\ \dot{\varepsilon}_2,\ \ldots,\ \dot{\varepsilon}_6)^T$$

$$u = (|f_1|,\ |f_2|,\ \ldots,|f_\alpha|)^T$$

$$\varepsilon_i = q_i(t)/l \qquad (i = 1, 2, \ldots, 6)$$

$$
B = \begin{bmatrix}
\dfrac{-l_{z1}\hat{f}_{y1}+l_{y1}\hat{f}_{z1}}{J_x^{(0)}\omega_c^2}, & \dfrac{-l_{z2}\hat{f}_{y2}+l_{y2}\hat{f}_{z2}}{J_x^{(0)}\omega_c^2}, & \ldots, & \dfrac{-l_{z\alpha}\hat{f}_{y\alpha}+l_{y\alpha}\hat{f}_{z\alpha}}{J_x^{(0)}\omega_c^2} \\
\dfrac{-l_{y1}\hat{f}_{x1}+l_{x1}\hat{f}_{y1}}{J_z^{(0)}\omega_c^2}, & \dfrac{-l_{y2}\hat{f}_{x2}+l_{x2}\hat{f}_{y2}}{J_z^{(0)}\omega_c^2}, & \ldots, & \dfrac{-l_{y\alpha}\hat{f}_{x\alpha}+l_{y\alpha}}{J_z^{(0)}\omega_c^2} \\
\dfrac{l_{z1}\hat{f}_{x1}-l_{x1}\hat{f}_{z1}}{J_y^{(0)}\omega_c^2}, & \dfrac{l_{z2}\hat{f}_{x2}-l_{x2}\hat{f}_{z2}}{J_y^{(0)}\omega_c^2}, & \ldots, & \dfrac{l_{z\alpha}\hat{f}_{x\alpha}-l_{x\alpha}\hat{f}_{z\alpha}}{J_y^{(0)}\omega_c^2} \\
\dfrac{\phi_x^{(1)}(\xi_1,\beta_1)}{M_1 l\omega_c^2}, & \dfrac{\phi_x^{(1)}(\xi_2,\beta_2)}{M_1 l\omega_c^2}, & \ldots, & \dfrac{\phi_x^{(1)}(\xi_\alpha,\beta_\alpha)}{M_1 l\omega_c^2} \\
\vdots & & & \\
\dfrac{\phi_x^{(6)}(\xi_1,\beta_1)}{M_6 l\omega_c^2}, & \dfrac{\phi_x^{(6)}(\xi_2,\beta_2)}{M_6 l\omega_c^2}, & \ldots, & \dfrac{\phi_x^{(6)}(\xi_\alpha,\beta_\alpha)}{M_6 l\omega_c^2}
\end{bmatrix}
$$

ψ, ϕ, θ yaw, roll and pitch angles, respectively, between the undeformed axis of the shallow spherical shell

$q_i(t)$ modal amplitude of the i^{th} generic mode whose shape function is $\phi^{(i)}$

ω_c orbital angular rate, constant for assumed circular orbit

l characteristic length (the base radius)

M_x i^{th} modal mass

$\phi_x^{(i)}$ the x-axis component of the i^{th} modal shape function

l_{xi}, l_{yi}, l_{zi} the components of the moment arm for the i^{th} actuator

Eq. (58) is nondimensionalized according to τ.

$$\tau = \omega_c t,\ \varepsilon_i = q_i(t)/l \qquad (i = 1,\ 2,\ \ldots,\ 6)$$

The derivative in Eq. (58) is with respect to τ.

For all of these cases it is assumed that two Earth horizon sensors are used for measuring: (1) the angle between the shell's roll axis (nominally along the local horizontal) and the local vertical axis and (2) the angle between the shell's pitch axis (nominally along the orbit normal) and the local vertical axis.

It is also assumed that two Sun sensors are used for measuring the angle between the shell's roll axis and the direction vector of the Sun, and the angle between the shell's pitch axis and the direction vector of the Sun.

Finally, it is assumed that the six displacement sensors are used to measure the shell's transverse displacement parallel to the shell's yaw axis. For each case analyzed in Table 2, the displacement sensors are considered to be colocated with the actuators.

The mathematical equations of the observational model can be found in [9].

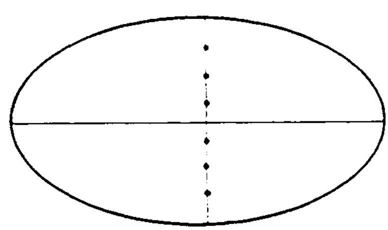
It is assumed that the optimal LQG digital controllers with predicting observer are used to control the orientation and shape of the shallow spherical shell and the number of actuators is six. The arrangement of the six actuators may be assumed for the following seven cases (Table 2):

Table 2. The Actuator Locations (ξ, β) and Force Directions $(\hat{f}_x\ \hat{f}_y\ \hat{f}_z)$

CASE 1

Actuator No.	ξ	β	f_x	f_y	f_z
1	0.28	0°	1	0	0
2	0.57	0°	1	1	0
3	0.84	0°	1	1	1
4	0.28	180°	1	0	0
5	0.57	180°	1	1	0
6	0.84	180°	1	1	1

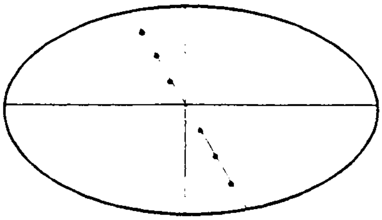
Location of Actuator



CASE 2

Actuator No.	ξ	ß	f_x	f_y	f_z
1	0.28	5°	1	0	0
2	0.57	5°	1	1	0
3	0.84	5°	1	1	1
4	0.28	185°	1	0	0
5	0.57	185°	1	1	0
6	0.84	185°	1	1	1

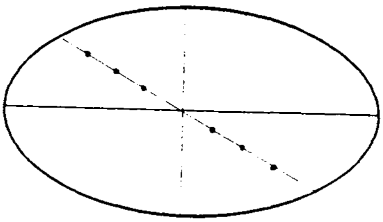
Location of Actuator



CASE 3

Actuator No.	ξ	ß	f_x	f_y	f_z
1	0.28	45°	1	0	0
2	0.57	45°	1	1	0
3	0.84	45°	1	1	1
4	0.28	225°	1	0	0
5	0.57	225°	1	1	0
6	0.84	225°	1	1	1

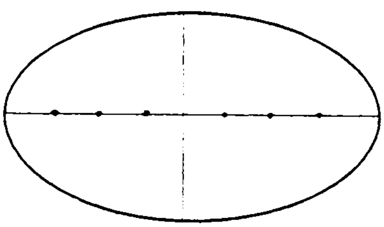
Location of Actuator



CASE 4

Actuator No.	ξ	ß	f_x	f_y	f_z
1	0.28	90°	1	0	0
2	0.57	90°	1	1	0
3	0.84	90°	1	1	1
4	0.28	270°	1	0	0
5	0.57	270°	1	1	0
6	0.84	270°	1	1	1

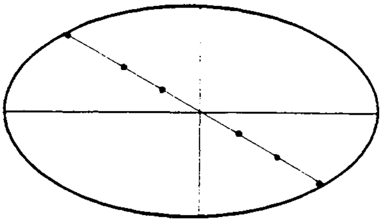
Location of Actuator



CASE 5

Actuator No.	ξ	β	f_x	f_y	f_z
1	0.28	45°	1	0	0
2	0.57	45°	1	0	0
3	1.00	45°	0	-sin45°	cos45°
4	0.28	225°	1	0	0
5	0.57	225°	1	0	0
6	1.00	225°	0	sin45°	-cos45°

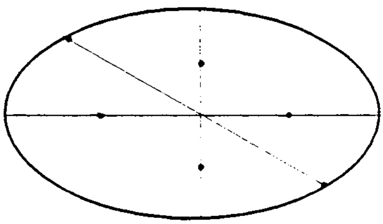
Location of Actuator



CASE 6

Actuator No.	ξ	β	f_x	f_y	f_z
1	0.57	0°	1	0	0
2	0.57	90°	1	0	0
3	0.57	180°	0	0	0
4	0.57	270°	1	0	0
5	1.00	45°	0	-sin45°	cos45°
6	1.00	225°	0	sin45°	cos45°

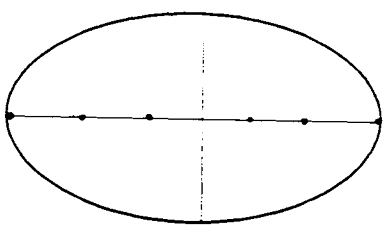
Location of Actuator



CASE 7

Actuator No.	ξ	β	f_x	f_y	f_z
1	0.28	90°	1	0	0
2	0.57	90°	1	0	0
3	1.00	90°	1	-1	0
4	0.28	270°	1	0	0
5	0.57	270°	1	0	0
6	1.00	270°	0	1	0

Location of Actuator

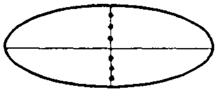
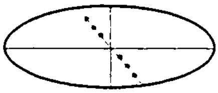
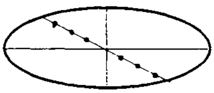
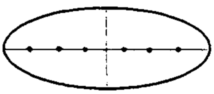
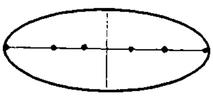
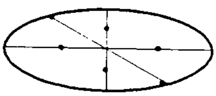
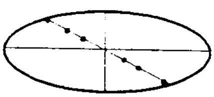


The seven cases may be divided into two groups. The first group includes case 1 through case 4. The second group includes case 5 through case 7. In the first group there are six jets (actuators) for each case. In the second group there are six jets for each case, but two of the jets for the second group are located at the edge of the shell with their thrust direction tangent to the shell's (circular) edge. In order to reduce the possibility that the jets used primarily for shape control would also disturb the orientation of the shell, the placements of these jets are arranged symmetrically with respect to the shell's undeformed principal axes.

We have three kinds of different measures for the degree of controllability for each case. It should be pointed out that the three scalar measures should be used at the same time. Generally speaking, if the system "1" is more controllable than the system "2", the value of every one of the μ_i (i = 1, 2, 3) for the system "1" should be greater than the corresponding value of the μ_i (i = 1, 2, 2) for the system "2". Because the degree of controllability (observability) is itself a property of the system, it is independent of the initial state of the system. This fact indicates in geometrical terms that the super-ellipsoid of equi-control effort for the system "1" contains the super-ellipsoid of the equi-control effort for the system "2". Sometimes the relationships for the μ_i (i = 1, 2, 3) between the system "1" and the system "2" are not consistent; for example, if the values of two of the μ_i (i = 1, 2, 3) for the system "1" are greater than those of the system "2", and the value of one of the μ_i (i = 1, 2, 3) for the system "1" is less than the corresponding one of the μ_i (i = 1, 2, 3) for the system "2", then the system "1" still can be taken as the more controllable system.

The values of the degree of controllability for each case are listed in Table 3. The reason why the degree of controllability for case 1 is zero is that all the actuators are located along the meridional nodal line of one of the fundamental shell vibrational modes.

Table 3. The Degree of Controllability for the Different Cases

Case	Location of Actuator	μ_1	μ_2	μ_3
1		0.0	0.0	0.0
2		$0.13875\text{x}10^{-12}$	$0.17656\text{x}10^{-11}$	$0.67227\text{x}10^{-7}$
3		$0.17673\text{x}10^{-11}$	$23696\text{x}10^{-10}$	$0.25045\text{x}10^{-6}$
4		$0.17974\text{x}10^{-11}$	$0.26434\text{x}10^{-10}$	$0.31558\text{x}10^{-6}$
5		$0.14332\text{x}10^{-10}$	$0.12421\text{x}10^{-9}$	$0.70448\text{x}10^{-6}$
6		$0.48360\text{x}10^{-12}$	$0.46915\text{x}10^{-11}$	$0.61384\text{x}10^{-6}$
7		$0.14332\text{x}10^{-10}$	$0.14788\text{x}10^{-9}$	$0.10084\text{x}10^{-5}$

In the first group of actuators, the best placement of actuators for which the degree of controllability is highest is case 4. In the second group, the best placement of actuators is case 7, and the degree of controllability for case 7 is also higher than that for case 4. The reason why the degree of controllability for case 6 is so low is that four actuators are all located on the same modal circle (ξ constant).

As far as the comparison between case 6 and cases 3 and 4 are concerned, we should select case 4 as the best one among the three cases, since two of the μ_i ($i = 1, 2, 3$) for case 4 are clearly larger than the corresponding μ_i for either case 3 or case 6.

In summary, the locations of the actuators should be as far as possible from the nodal lines, and also should be arranged so that as few actuators as possible will be located on the same nodal circles.

In order to control the orientation of the shell effectively, a combination of tangential jets along the edge of the shell together with selected jets normal to the shell's major surface is recommended.

If the number of actuators is limited to six, the arrangement of actuators as in Case 7 is suggested.

The transient responses with the LQG control for case 2 and case 7 are selected to show the differences in the attitude and the first three modal amplitude responses.

The initial conditions are selected arbitrarily and are the same for the two cases, i.e., roll $\phi(0)=0.1$ rad., yaw $\psi(0)=0.1$ rad., pitch $\theta(0)=0.1$ rad. The initial conditions for the first six modal amplitudes are $q_1(0) = 1$ meter, $q_2(0) = q_3(0) = \ldots$ $q_6(0) = 0.0$, respectively.

By selecting an initial displacement in one of the modal amplitude (in these cases, the first modal amplitude) it is possible to demonstrate the excitation of the other modes due to coupling.

The transient response of the attitude motion for case 2 is shown in Fig. 1, the transient responses of the 1st - 3rd modal amplitudes for case 2 are shown in Fig. 2. The transient responses for the attitude motion and the modal amplitudes for case 7 are shown in Fig. 3 and Fig. 4, respectively. The comparison of the first modal amplitude responses for case 2 and case 7 is shown in Fig. 5.

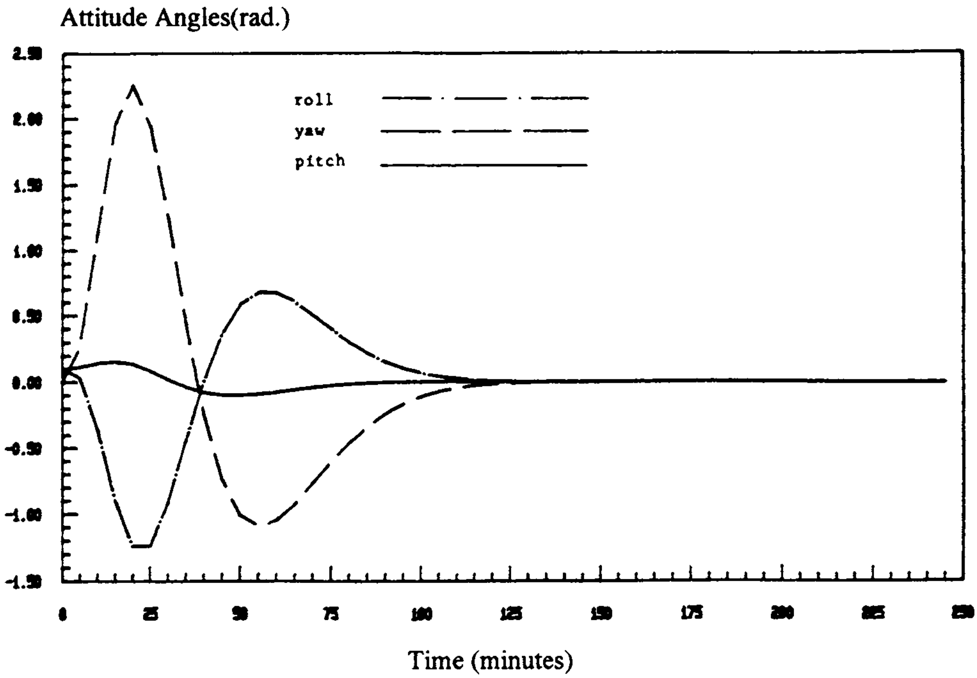


Fig.1 Attitude response of the shallow shell LQG control for case 2

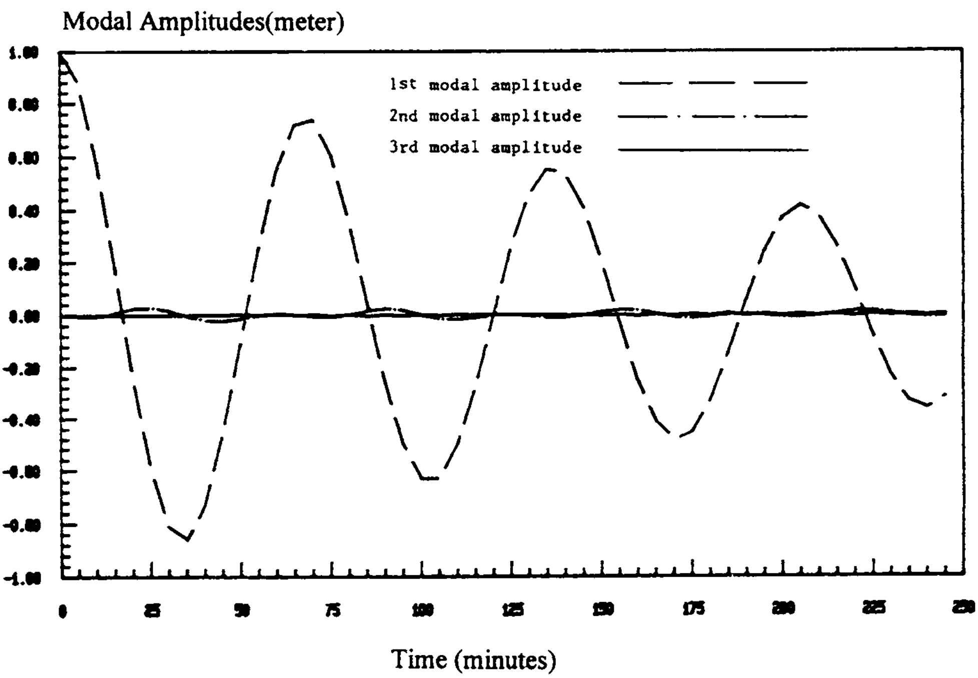


Fig.2 Modal amplitude response of the shallow shell LQG control for case 2

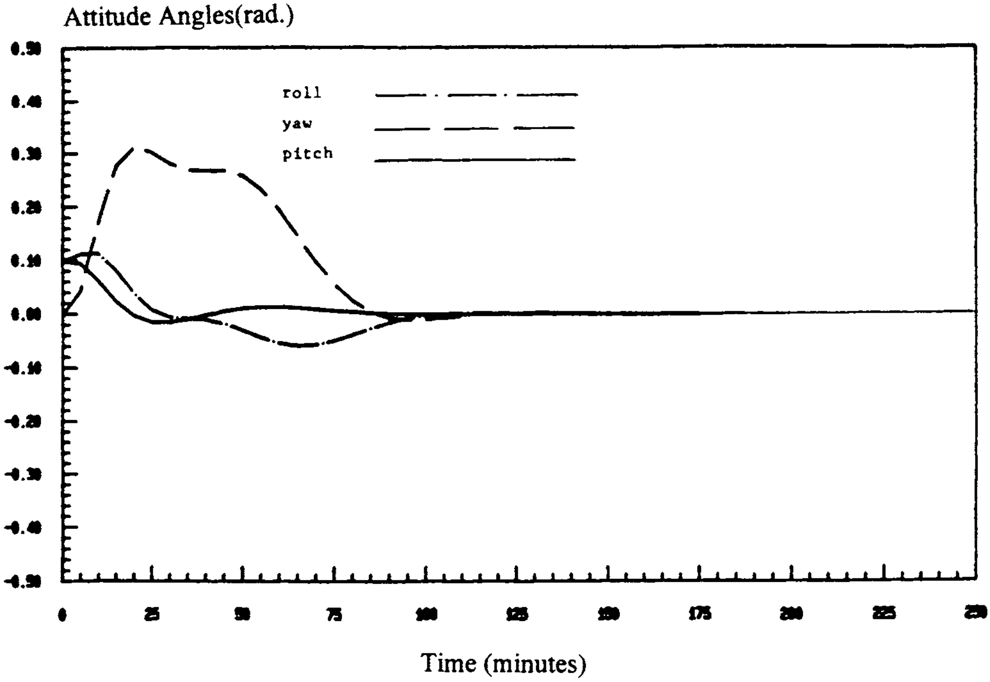


Fig.3 Attitude response of the shallow shell LQG control for case 7

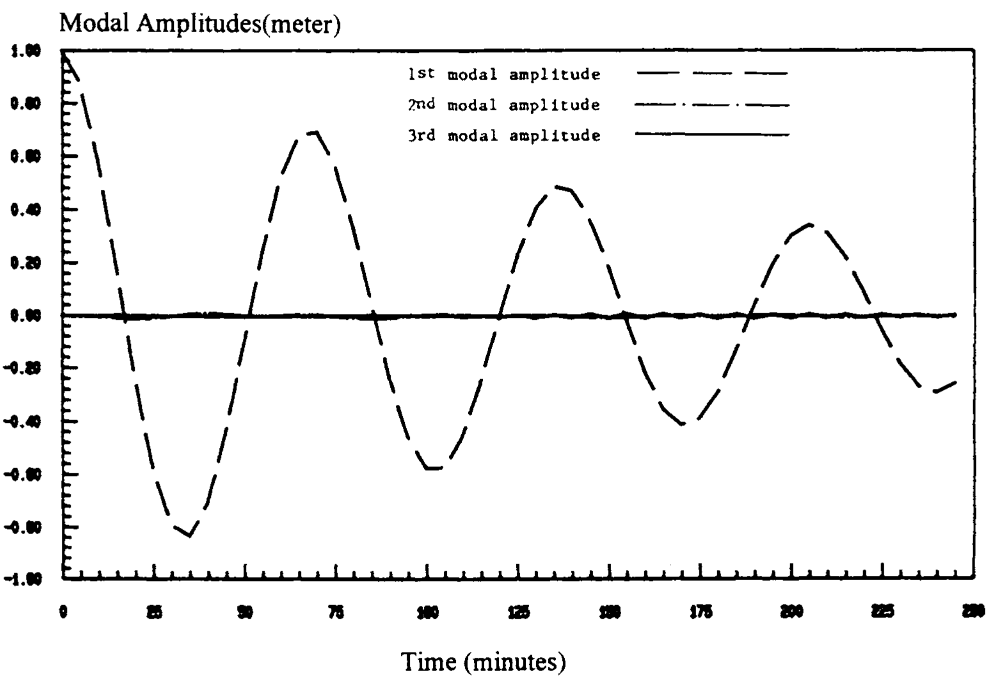


Fig.4 Modal amplitude response of the shallow shell LQG control for case 7

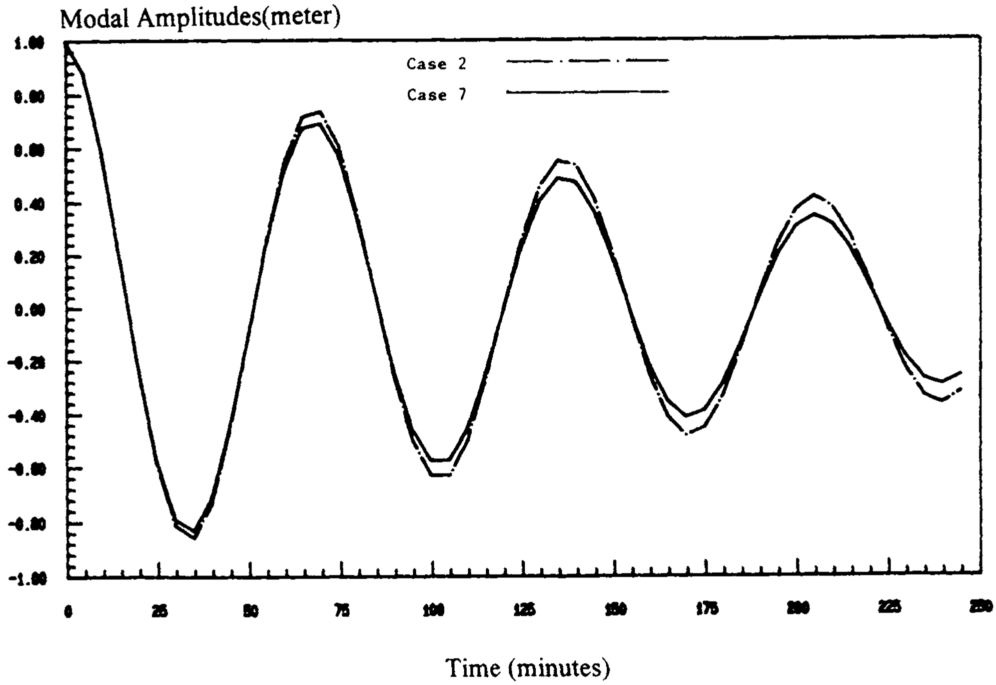


Fig.5 The first modal amplitudes responses for case 2 and case 7

It is shown in Figs. 1-5 that the roll is coupled strongly with yaw, but the coupling of roll, yaw with pitch is very weak. The quality of the transient responses of case 2 is much worse than that of case 7. This is a result of the differences in the degree of controllability for case 2 and case 7.

VII. CONCLUSIONS

Three candidate definitions of the degree of controllability and observability are presented for linear discrete-time systems based on the scalar measure of the Grammian matrix. Their general properties, together with the physical and geometrical interpretations for the fuel optimal control problem are shown in detail. The advantages of these kinds of definitions for the degree of controllability (observability) are the clarity of the physical and geometrical interpretations and the simplicity of the resulting calculation. Thus, they are very useful for practical engineering design.

The transient responses for several typical systems with different degrees of controllability show that the quality of the system transient response depends on its degree of controllability: transient responses of the system with a higher degree of controllability are better than those of the system with a lower degree of controllability. The applications of the concept of the degree of controllability for actuator placement of the orbiting shallow spherical shell system are successfully implemented in this paper.

ACKNOWLEDGEMENTS

This research was supported by the NASA/Howard University Large Space Structures Institute (LSSI) and AFWAL Contract No. F33615-88C-3208. Appreciation is extended to Dr. Jerrold Housner, NASA Langley LSSI Technical Director, Dr. Jer-Nan Juang, NASA Technical Project Monitor, and Dr. Taft Broome, Jr., Howard University LSSI Director. Also thanks are extended to Dr. V.B. Venkayya, AFWAL/ FIBRA Project Engineer and Lt. Bruce Snyder, AFWAL.

REFERENCES

1. Kalman, R.E., Ho, Y.C. and Narendra, K.S., "Controllability of Linear Dynamical Systems," Contributions to Differential Equations, Vol. 1, No. 2, 1961, pp. 182-213.
2. Tomovic, R., "Controllability, Invariancy, and Sensitivity," Proc. Third Allerton Conference, 1965, pp. 17-26.
3. Johnson, C.D., "Optimization of a Certain Quality of Complete Controllability and Observability for Linear Dynamical Systems," A.S.M.E. Trans. J. Basic Engrg., Vol. 191, Ser. D, 1969, pp. 228-238.
4. Muller, P.C. and Weber, H.I., "Analysis and Optimization of Certain Qualities of Controllability and Observability for Linear Dynamical

Systems," Automatica, Vol. 8, 1972, pp. 237-246.

5. Viswanathan, C.N., Longman, R.W. and Likins, P.W., "A Definition of the Degree of Controllability - A Criterion for Actuator Placement," Proc. of Second VPI&SU/AIAA Symposium on Dynamics and Control of Large Flexible Spacecraft, Blacksburg, VA, June 1979, pp. 369-381.

6. Xing Guangqian and Bainum, Peter M., "The Equations of Motion for a General Orbiting Large Space Flexible System," Presented at the Sixteenth International Symposium on Space Technology and Science, May 22-27, 1988, Sapporo, Hokkaido, Japan.

7. Marshall, A.W. and Olkin, I., Inequalities: Theory of Majorization and Its Applications, Academic Press, 1979, p. 478, p. 475.

8. Mitrinovic, D.S., Analytical Inequalities, Springer-Verlag, New York: Heidelberg. Berlin, 1970, p. 55.

9. Xing, Guangqian and Bainum, Peter M., "The Optimal LQG Digital Shape and Orientation Control of an Orbiting Shallow Spherical Shell System," the 40th Congress of the International Astronautical Federation, Malaga, Spain, 7-13 October 1989; see also Acta Astronautica, Vol. 21, No. 10, pp. 719-731, 1990.

APPENDIX

1. The Proof of Invariability of Degree of Controllability under Orthogonal Linear Transformation

It is supposed that the linear discrete-time system is as follows:

$$X_k = \Phi X_{k-1} + \Gamma u_{k-1} \tag{A1}$$

let

$$X_k = L\bar{x}_k \tag{A2}$$

where

L = orthogonal linear transformation

The equation of the new system after linear transformation is as follows:

$$\bar{x}_k = \bar{\Phi}\bar{x}_{k-1} + \bar{\Gamma}u_{k-1} \tag{A3}$$

where

$$\bar{\Phi} = L^{-1}\Phi L = L^{T}\Phi L, \quad L^{-1}\Gamma = L^{T}\Gamma = \bar{\Gamma}$$

The controllability matrix for the system (A1) is

$$P_c = (\Gamma \,\vdots\, \Phi\Gamma \,\vdots\, \Phi^2\Gamma \,\vdots\, \dots \, \Phi^{n-1}\Gamma) \tag{A4}$$

The degree of controllability matrix W_c is

$$W_c = P_c P_c^T = \Gamma\Gamma^T + \Phi\Gamma\Gamma^T\Phi^T + \dots + \Phi^{n-1}\Gamma\Gamma^T(\Phi^{n-1})^T$$

The controllability matrix for the system (A3) W_c is as follows

$$\begin{aligned}\bar{P}_c &= (\bar{\Gamma} \,\vdots\, \bar{\Phi}\,\bar{\Gamma} \,\vdots\, \bar{\Phi}^2\bar{\Gamma} \,\vdots\, \dots \,\vdots\, \bar{\Phi}^{n-1}\bar{\Gamma}) \\ &= (L^{-1}\Gamma \,\vdots\, L^{-1}\Phi L L^{-1}\Gamma \,\vdots\, \dots \,\vdots \\ &\quad (L^{-1}\Phi L)(L^{-1}\Phi L) \dots (L^{-1}\Phi L)L^{-1}\Gamma) \\ &= L^{-1}(\Gamma \,\vdots\, \Phi\Gamma \,\vdots\, \dots \,\vdots\, \Phi^{n-1}\Gamma) = L^{-1}P_c\end{aligned}$$

The degree of controllability matrix for the system (A3), W_c, is as follows:

$$\overline{W}_c = \overline{P}_c \overline{P}_c^T = L^{-1} P_c P_c^T L \tag{A5}$$

As we know, the eigenvalues of matrix are invariant under a similarity transformation, i.e.,

$$\Lambda(\overline{W}_c) = \Lambda(L^{-1} P_c P_c^T L) = \Lambda(P_c P_c^T) = \Lambda(W_c) \tag{A6}$$

Therefore, the degree of controllability is also invariant under a similarity transformation, i.e.,

$$\mu_i(\overline{W}_c) = \mu_i(W_c) \quad (i = 1,\ 2,\ 3) \tag{A7}$$

2. The Proof of the General Properties (a) - (d) for the Degree of Controllability $\mu_i(W_c)$ (i = 1, 2, 3)

(1) The case for μ_i

It is evident that the properties (a) - (c) for μ_i are true. What we need to prove is property (d). That the property (d) is true is due to the following facts.

If $W_c = W_{c1} + W_{c2}$; W_{c1}, $W_{c2} \epsilon W_c^*$, then

$$\mu_1(W_c) = \lambda_{\min}(W_c) = \underset{\substack{\|x\|=1 \\ x\in R^n}}{Min} <W_c x, x> = \underset{\|x\|=1}{Min}\{<W_{c1}x, x>$$

$$+ <W_{c2}x, x>\} \geq \underset{\|x\|=1}{Min} <W_{c1}x, x> + \underset{\|x\|=1}{Min} \{<W_{c2}x, x> \tag{A8}$$

$$= \lambda_{\min}(W_{c1}x) + \lambda_{\min}(W_{c2}) = \mu_1(W_{c1}) + \mu_1(W_{c2})$$

(2) The case for μ_2

Because the trace of a positive definite symmetrical matrix is invariant under the similarity transformation, it is evident that the properties (a) - (c) are true. The proof of property (d) for μ_2 is as follows.

Based on the definition of μ_2, what we want to prove is the following inequality:

$$\mu_2(W_c) = \frac{n}{trW_c^{-1}} \geq \frac{n}{trW_{c1}^{-1}} + \frac{n}{trW_{c2}^{-1}} \tag{A9}$$
$$= \mu_2(W_{c1}) + \mu_2(W_{c2})$$

where $W_{c1}, W_{c2} \in W_c^*$, $W_c=W_{c1} + W_{c2}$

It is well known that

$$W_c^{-1} = Adj\, W_c/\det W_c$$

then

$$trW_c^{-1} = \sum_{i=1}^{n} detW_c(i)/\det W_c \tag{A10}$$

where $detW_c(i)$ is the determinate of the ith principal minor of the matrix W_c. Considering Eq. (A10), the inequality (A9) we want to prove will become

$$\frac{n}{\sum_{i=1}^{n} \frac{\det(W_{c1}(i)+W_{c2}(i))}{\det(W_{c1}+W_{c2})}} \geq \frac{n}{\sum_{i=1}^{n} \frac{detW_{c1}(i)}{\det W_{c1}}} + \frac{n}{\sum_{i=1}^{n} \frac{detW_{c2}(i)}{\det W_{c2}(i)}} \tag{A11}$$

From the well known Berstrom inequality[7], we have

$$\frac{\det(W_{c1}+W_{c2})}{\det(W_{c1}(i)+W_{c2}(i))} \geq \frac{detW_{c1}}{detW_{c1}(i)} + \frac{detW_{c2}}{\det W_{c2}(i)} \tag{A12}$$

From (A12), we have

$$\frac{\det(W_{c1}(i)+W_{c2}(i))}{\det(W_{c1}+W_{c2})} \leq \frac{1}{\frac{detW_{c1}}{detW_{c1}(i)} + \frac{detW_{c2}}{detW_{c2}(i)}} \tag{A13}$$

then we also have

$$\sum_{i=1}^{n} \frac{\det(W_{c1}(i) + W_{c2}(i))}{\det(W_{c1}+W_{c2})} \leq \sum_{i=1}^{n} \left(\frac{\det W_{c1}}{\det W_{c1}(i)} + \frac{\det W_{c2}}{\det W_{c2}(i)} \right)^{-1} \quad \text{(A14)}$$

i.e.,

$$\left[\sum_{i=1}^{n} \frac{\det(W_{c1}(i)+W_{c2}(i))}{\det(W_{c1}+W_{c2})} \right]^{-1} \geq \left[\sum_{i=1}^{n} \left(\frac{\det(W_{c1})}{\det W_{c1}(i)} + \frac{\det W_{c2}}{\det W_{c2}(i)} \right)^{-1} \right]^{-1} \quad \text{(A15)}$$

Because the $\det W_{c1}/\det W_{c1}(i)$, $\det W_{c2}/\det W_{c2}(i)$ are all real numbers, by applying the Minkovski inequality [8]

$$\left[\sum_{i=1}^{n} \left[\frac{\det W_{c1}}{\det(W_{c1}(i)} + \frac{\det W_{c2}}{\det W_{c2}(i)} \right]^{-1} \right]^{-1}$$

$$\geq \left[\sum_{i=1}^{n} \left[\frac{\det W_{c1}}{\det(W_{c1}(i)} \right]^{-1} \right]^{-1} + \left[\sum_{i=1}^{n} \left[\frac{\det W_{c2}}{\det W_{c2}(i)} \right]^{-1} \right]^{-1} \quad \text{(A16)}$$

$$= \left[\sum_{i=1}^{n} \frac{\det(W_{c1}(i)}{\det W_{c1}} \right]^{-1} + \left[\sum_{i=1}^{n} \frac{\det W_{c2}(i)}{\det W_{c2}} \right]^{-1}$$

Substituting (A16) into inequality (A15), we have

$$\left[\sum_{i=1}^{n} \frac{\det(W_{c1}(i)+W_{c2}(i))}{\det(W_{c1}+W_{c2})}\right]^{-1} \geq$$

$$\left[\sum_{i=1}^{n} \frac{\det W_{c1}(i)}{\det W_{c1}}\right]^{-1} + \left[\sum_{i=1}^{n} \frac{\det W_{c2}(i)}{\det W_{c2}}\right]^{-1} \tag{A17}$$

If the two sides of inequality (A17) are multiplied by n, the result is just what we need, (A11).

(3) The case for μ_3

It is evident that the properties (a) - (c) for μ_3 are true due directly to the definition of μ_3. Based on the Oppenheim inequality [7], we have

$$(\det(W_{c1}+W_{c2}))^{\frac{1}{n}} \geq (\det W_{c1})^{\frac{1}{n}} + (\det W_{c2})^{\frac{1}{n}} \tag{A18}$$

This is just what we want to show property (d) for μ_3, i.e.,

$$\mu_3(W_{c1}+W_{c2}) \geq \mu_3(W_{c1}) + \mu_3(W_{c2}) \tag{A19}$$

TECHNIQUES IN RECONFIGURABLE CONTROL SYSTEM DESIGN

Zhiqiang Gao

Department of Electrical Engineering
Cleveland State University
Cleveland, Ohio 44115
U.S.A

I. INTRODUCTION

The reconfigurable control systems (RCS) are control systems that possess the ability to accommodate system failures automatically based upon a-priori assumed conditions. It was also known as restructurable, or self-repairing, or failure tolerant control systems. The research in this area was initially motivated by the control problèms encountered in the aircraft control system design in the early 80's. In that case, the ideal goal is to achieve the so called "fault-tolerant", or, "self-repairing" capability in the flight control systems, so that unanticipated failures in the system can be accommodated and the impaired airplane can be, at least, landed safely whenever possible. Over the years, the nature of the research has become that of automatic failure accommodation. The problems addressed here are different from standard control problems in that, first, there is usually a severe time constraint associated with failure accommodations which require that the control systems be adjusted in real-time without much human intervention; secondly, the failures in the physical systems often result in drastic changes in system dynamics which make it almost impossible to be handled by one controller, even when the controller is deemed as "robust".

One may view control reconfiguration in the face of failures as a two step process. First, the system must be stabilized before it disintegrates. The way to

CONTROL AND DYNAMICS SYSTEMS, VOL. 79

achieve it is very much system dependent. For a stable plant, a conservative low gain compensator can usually be found to do the job. For an unstable plant, a stabilizing compensator must be selected from a predetermined set or be determined quickly on-line based on the available information on failures. It is unrealistic to assume that the failure detection and identification (FDI), the determination of the new parameters of the impaired system, and the design and computation of the stabilizing controller all be carried out in real-time, given the limitations in all three areas plus the demand on the computation resources. Thus, instead of trying to accommodate all possible failures, a more practical RCS objective is to deal with failures that can be quickly brought under control, at least temporarily.

The success of the first step will give valuable time for the second step, which is to redesign the controller to recover performance and robustness as much as possible. The achievable performance is inverse proportional to the amount of uncertainties. It takes time for the FDI and the system identification algorithms to provide accurate information needed in control reconfiguration.

The control reconfiguration approach must be algorithmic, that is, it must be implementable in a computer algorithm. Most current control design methods, however, such as Linear Quadratic Regulator (LQR), H_∞, Eigenstructure Assignment, to name a few, require at least a few iterations, such as selections of weighting functions, etc., before arriving at the final design. Unfortunately, such iterative process is not feasible because of the on-line nature and the time constraints in RCS design. This must be addressed in any proposed method.

In this chapter, a few techniques proposed for RCS design will be reviewed. The Pseudo-Inverse Method and its extensions are discussed in Section II. The Linear model-following based approaches are demonstrated in Section III. Other methods such as the LQ and Eigenstructure Assignment approaches are briefly introduced in Section IV. Finally, some concluding remarks can be found in Section V.

II. THE PSEUDO-INVERSE METHOD

The Pseudo-Inverse Method(PIM), is a key approach to reconfigurable control and it has been successfully implemented in flight simulations [3,24,36,38]. The main idea is to modify the feedback gain so that the reconfigured system approximates the nominal system in some sense. It is attractive because of its simplicity in computation and implementation.

The main objective of the PIM is to maintain as much similarity as possible to the original design and thus to provide graceful degradation in performance. This is achieved by reassigning the feedback gain as illustrated below: let the open-loop plant be given as

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.1}$$

where $A \in R^{nxn}$, $B \in R^{nxm}$ and $C \in R^{qxn}$. Assume that the nominal closed-loop system is designed by using the state feedback $u = Kx$, $K \in R^{mxn}$, and the closed-loop system is

$$\begin{aligned} \dot{x} &= (A+BK)x \\ y &= Cx \end{aligned} \tag{2.2}$$

where K is the state feedback gain. Suppose that the model of the system, in which failures have occurred, is given as

$$\begin{aligned} \dot{x}_f &= A_f x_f + B_f u_f \\ y_f &= C_f x_f \end{aligned} \tag{2.3}$$

where $A_f \in R^{nxn}$, $B_f \in R^{nxm}$ and $C_f \in R^{qxn}$, and the new closed-loop system is

$$\dot{x}_f = (A_f+B_fK_f)x_f$$

$$y_f = C_f x_f \tag{2.4}$$

where K_f is the new feedback gain to be determined. In the PIM[36], the objective is to find a K_f so that the closed-loop transition matrix in (2.4) approximates in some sense the one in (2.2). For this, A+BK is equated to $A_f+B_fK_f$ and an approximate solution for K_f is given by

$$K_f = B_f^+(A - A_f + BK) \tag{2.5}$$

where B_f^+ denotes the pseudo-inverse of B_f.

Note that K_f can be calculated from (2.5) for many anticipated failures and be stored in the flight control computer. Once the failure has occurred and is identified, that is the model of the system with failure (2.3) is obtained, the feedback gain can then be modified. This is considered as a relatively fast solution to stabilize the impaired airplane[36]. This PIM method has also been used for on-line accommodation for unanticipated failures in various forms [2,24,36,38]. There is one problem, however, which might render the method useless, namely, that the solution from (2.5) does not necessarily make the closed-loop system in (2.4) stable. This issue will be addressed in the modified PIM in Section 2.2.

2.1 The Properties of the PIM [12]

Although the PIM has been used widely in the study of the RCS, it is still an ad hoc approach. The theoretical aspects of this methods were not fully investigated at the beginning. For example, it was not clear in what sense the closed-loop system in (2.4) approximates the one in (2.2) when K_f is obtained from (2.5). In the PIM, it is desirable to have

$$A + BK = A_f + B_fK_f \tag{2.6}$$

This equation may or may not have an exact solution depending on the row rank of the matrix B_f. If B_f has full row rank, then (2.5) always satisfies (2.6); otherwise, there is no exact solution to (2.6) and K_f from (2.5) is only an approximate solution. It is interesting to see what the PIM implies in terms of the eigenvalues, which are often used in the specifications of the performance criteria.

Lemma 2.1 Let

$$J = \| (A+BK) - (A_f+B_fK_f) \|_F \tag{2.7}$$

where $\|\cdot\|_F$ stands for the Frobenius norm. Then the K_f obtained from (2.5) minimizes J.

Lemma 2.1 shows that the solution (2.5), used in the PIM, makes the closed-loop system (2.4) approximate the nominal one (2.2) in the sense that the Frobenius norm of the difference of the A matrices is minimized. The underlining idea in the PIM is that if the norm J is minimized, hopefully the behavior of the reconfigured system will be close to that of the nominal system. It goes without saying that we would like to know just how close it is going to be and in what sense exactly. The relation between variations of closed-loop eigenvalues and the Frobenius norm in (2.7) is given by the following theorem:

Theorem 2.1: Let (A+BK) be non-defective, that is, it can be reduced to diagonal form by a similarity transformation, and let X be the eigenvector matrix of A+BK in (2.2) and $X^{-1}(A+BK)X = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$. The eigenvalues of $A_f+B_fK_f$ in (2.4) lie in the union of the disks

$$D_i = \{ \lambda : | \lambda - \lambda_i | \leq \| X \|_2 \| X^{-1} \|_2 J \} \tag{2.9}$$

where J is defined in (2.7) and $\|\cdot\|_2$ denotes the matrix 2-norms.

The significance of this theorem is that it provides a bound to the variation of the eigenvalues in terms of eigenvectors of the nominal system and the Frobenius norm J. If the nominal system is robust in the sense that all the eigenvalues are relatively far left of the $j\omega$ axis, then the PIM has a good chance to work provided J is small enough; J can be made small when the failure is not "too severe". This fact must be taken into consideration in the design of the nominal system. Theorem 2.1 also shows the limitations of the PIM. It is clear that by just minimizing J, stability is not guaranteed; the minimum J can be large enough to allow the eigenvalues of the closed-loop system shift to the right half plane. That is, the use of the PIM in the automatic reconfiguring control system may result in an unstable closed-loop system unless it is restricted to only certain classes of failure where the value of J is small enough, in which case all the disks in (2.9) lie in the left half plane.

2.2 The Modified Pseudo-Inverse Method [12]

In the implementation of the RCS, the stability is probably the most important property of the system. Although the PIM is simple and easy to implement, the lack of stability guarantees puts constraints on its application. The objective of the modified PIM (MPIM) is to maintain closed-loop stability while recovering the performance as much as possible. Equivalently, the objective is to keep the system in (2.4) stable, while minimizing the norm in (2.7). Clearly this is a constraint minimization problem(CMP). In order to use standard, easy to implement, numerical algorithms on the CMP, the stability constraints must be expressed explicitly in terms of the individual elements of K_f as simple inequalities. To obtain such constraints, various methods are investigated in the area of robust stability of systems with parametric uncertainty.

In the analysis of system stability with parameter uncertainty in the state-space model, many approaches have been proposed to obtain bounds on the uncertainty that guarantee the stability of the perturbed system, see, for example, [17,42]. Zhou et al [42] gave an algorithm to compute the stability bounds on all the uncertain parameters in the state space. Gao et al [17] extended Zhou's results and obtained a much less conservative bound. These bounds can be directly used to derive stability constraints for parameters of K_f. Now, the RCS problem becomes that of minimizing the difference of the closed-loop 'A' matrices subject to the stability constraints, which is formally stated as follows:

Problem Formulation: Determine K_f to minimize J in (2.7) subject to the condition that (2.4) is stable (2.10)

Procedure:

Step1: Calculate K_f from (2.5);

Step2: Check the stability of (2.4) for the K_f obtained in step1;

Step3: If (2.4) is stable, stop; otherwise solve the constrained minimization defined in (2.10).

Remarks:

Detailed procedure for step 3 can be found in [12] where the stability bound on K_f is first determined and the closed-form solution of (2.10) is then derived. In implementing the MPIM in practice, the cases when the impaired systems are unstable must be considered. In these cases, the system has to be stabilized first by modifying the feedback gain. Many methods can be used here, but the key issue is that this process has to be computationally efficient. Only after the system is stabilized can the MPIM be applied to improve the performance of the system.

III. THE MODEL FOLLOWING APPROACH

The basic idea in the PIM and MPIM discussed above is to redesign the controller so that the impaired system is "close", in some sense, to the nominal system. These methods are attractive because of the simplicity in computation and implementation. This concept of designing a control system to mimic a desired model is obviously not new in control theory. Most notably, the early linear model following approach (LMF) [10], which is the basis of the modern model reference adaptive control (MRAC), employs a very similar concept. Several RCS design techniques were developed along this line of thought.

There is a definite link between the design objectives of the LMF and those of the PIM in terms of making one system, the plant or the impaired system, imitate another dynamic system, which is either a reference model (for LMF) or the nominal system (for PIM). The difference is that, in LMF, the plant approximates the reference model in terms of output trajectory, while in the PIM and MPIM the impaired system imitates the nominal system in terms of the closeness of the closed-loop 'A' matrices in their state space model. It is shown in [13] that the PIM is only a special case of the LMF.

A key problem in the LMF control system design is whether the plant can follow the reference model exactly, which is referred to as perfect model-following (PMF). PMF is desirable since it enables us to completely specify the behavior of a system. Without achieving PMF, the LMF approach can not guarantee how close the output of the plant is to that of the reference model and this arbitrariness may not be acceptable in many control applications such as the RCS. On the other hand, in conventional LMF, the conditions for PMF put severe constraints on the reference model which make it impractical. In this section, the classical LMF is first reviewed, followed by two recently proposed methods that address the PMF problem.

3.1. The Classical Linear Model Following Methods

LMF is a state-space design methodology by which a control system is designed to make the output of the plant follow the output of a model system with desired behavior. In this approach, the design objectives are incorporated into the reference model and the feedback and feedforward controllers are used, which are usually of zero order. By using a reference model to specify the design objectives, a difficulty in control system design is avoided, namely that the design specifications must be expressed directly in terms of the controller parameters. As in any control design approaches, there are limits in the attainable control specifications because of physical limitations and the allowable complexity of dynamic compensation. It is not always clear, however, how the system specifications should be chosen so that they are within those limits. In LMF, this is reflected in the constraints in the reference model for which the PMF can be achieved.

Assume that the plant and the reference model are of the same order. Let the reference model be given as:

$$\begin{aligned} \dot{x}_m &= A_m x_m + B_m u_m \\ y_m &= C_m x_m \end{aligned} \tag{3.1}$$

and the plant be represented by

$$\begin{aligned} \dot{x}_p &= A_p x_p + B_p u_p \\ y_p &= C_p x_p \end{aligned} \tag{3.2}$$

where x_m, $x_p \in R^n$, u_m, $u_p \in R^m$, A_m, $A_p \in R^{nxn}$, B_m, $B_p \in R^{nxm}$, C_m, $C_p \in R^{pxn}$. The corresponding transfer function matrices of the reference model and the plant are:

$$T_m(s) = C_m(sI - A_m)^{-1}B_m \tag{3.3}$$

$$P(s) = C_p(sI - A_p)^{-1}B_p. \tag{3.4}$$

Let e(t) represent the difference in the state variables,

$$e(t) = x_m(t) - x_p(t). \tag{3.5}$$

To achieve the PMF, one must insure that for any u_m, piecewise continuous, and $e(0) = 0$, we shall have $e(t) \equiv 0$ for all $t > 0$.

Next, we will discuss under what conditions the PMF is possible and how to find the feedback and feedforward controllers to achieve the PMF. In the cases when the PMF cannot be achieved, it is shown how the error can be minimized.

3.1.1 The Implicit LMF

In the control system configuration of implicit LMF, Figure 1, the reference model does not appear explicitly. Instead, the model is used to obtain the control parameters, k_u and k_p. From (3.1) and (3.2), by simple manipulation we have

$$\dot{e} = A_m e + (A_m - A_p)x_p + B_m u_m - B_p u_p \tag{3.6}$$

From the control configuration in Figure 1, the control input u_p has the form of

$$u_p = k_p x_p + k_u u_m \tag{3.7}$$

The PMF is achieved if the control parameter k_u and k_p are chosen such that

$$\dot{e} = A_m e \tag{3.8}$$

or equivalently

$$(A_m - A_p)x_p + B_m u_m - B_p u_p \equiv 0 \tag{3.9}$$

Note that if a solution u_p of (3.9) exists, it will take the form

$$u_p = B_p^+(A_m - A_p)x_p + B_p^+ B_m u_m \tag{3.10}$$

where B_p^+ represents the pseudo-inverse of the matrix B_p. From (3.10) k_u and k_p can be found as $k_p = B_p^+(A_m - A_p)$, and $k_u = B_p^+ B_m$. By substituting (3.10) in (3.6), a sufficient condition for the existence of the solution of (3.9) is

$$\begin{cases} (I - B_p B_p^+)(A_m - A_p) = 0 \\ (I - B_p B_p^+)B_m = 0 \end{cases} \tag{3.11}$$

Note that (3.11) is known as Erzberger's condition [10]. Clearly, these are rather restrictive conditions, since most systems have more states than inputs, $B_p B_p^+ \neq I$. Thus (3.11) can only be fulfilled when $(I - B_p B_p^+)$ is in both the left null spaces of $(A_m - A_p)$ and B_m. It seems for an arbitrary plant, it is rather difficult to find an appropriate reference model such that it represents the desired dynamics and, at the same time, satisfies (3.11).

It should be noted that even when the conditions for the PMF in (3.11) are not fulfilled, the solution in (3.10) still minimizes the 2-norm of the last three terms in the right side of (3.6). i.e.

$$\|(A_m-A_p)x_p + B_m u_m - B_p u_p\|_2 = \|\dot{e} - A_m e\|_2.$$

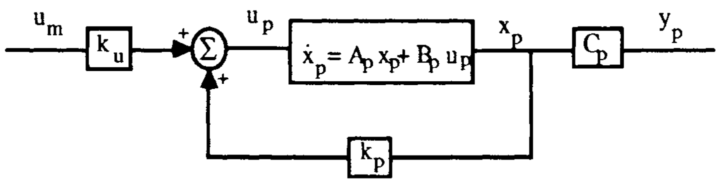


Figure 1 The control configuration of the implicit LMF.

This particular method of choosing u_p has the advantage of not involving x_m in the feedback thus eliminating the need for running the model on-line. Therefore the complexity of the control system is relatively low. One of the disadvantages of this method is that when the PMF is not achievable, the trajectory of *e* may not be desirable since we don't have control over the location of the poles in the system. Another disadvantage with this approach is that, when conditions in (3.11) are not satisfied, the solution (3.10) may result in an unstable system.

3.1.2 The Explicit LMF

A typical configuration of the explicit LMF is as illustrated in Figure 2. In this configuration, the reference model is actually implemented as part of the controller. To compare with the implicit LMF, let $e = x_m - x_p$, or equivalently, assume $C_m = I$ and $C_p = I$. By manipulating (3.1) and (3.2), $\dot{e}$ can also be written as

$$\dot{e} = A_p e + (A_m - A_p)x_m + B_m u_m - B_p u_p. \tag{3.12}$$

Let the control input be

$$u_p = u_1 + u_2 = (k_e e) + (k_m x_m + k_u u_m) \tag{3.13}$$

where $u_1 = k_e e$ is the stabilizing gain, and $u_2 = k_m x_m + k_u u_m$ is to be determined to minimize

$$\|(A_m - A_p)x_m + B_m u_m - B_p u_2\|_2 = \|\dot{e} - (A_p - B_p k_e)e\|_2.$$

From (3.12), it can be easily shown that the sufficient conditions for the PMF is exactly the same as in (3.11) and the corresponding control gains are:

$$k_m = B_p^+(A_m - A_p)$$

$$k_u = B_p^+ B_m$$

with k_e any stabilizing gain. Substituting (3.13) in (3.12), the equation of error is

$$\dot{e} = (A_p - B_p k_e)e + (I - B_p B_p^+)(A_m - A_p)x_m + (I - B_p B_p^+)B_m u_m. \tag{3.14}$$

When the conditions in (3.11) are met, we have

$$\dot{e} = (A_p - B_p k_e)e \tag{3.15}$$

In this approach if the plant is stabilizable, we can guarantee the stability of the closed-loop system by choosing k_e appropriately, regardless of whether the conditions in (3.11) are met or not. This can be illustrated as follows. Since $\{A_p, B_p\}$ is stabilizable, k_e can be chosen such that $(A_p - B_p k_e)$ has all its eigenvalues in the left half plane; furthermore, let $f(t) = (I - B_p B_p^+)(A_m - A_p)x_m(t) + (I - B_p B_p^+)B_m u_m(t)$, then (3.14) can be expressed as: $\dot{e} = (A_p - B_p k_e)e + f(t)$. Because the reference model in (3.1) is stable, x_m will be bounded and therefore f(t) will be bounded for any bounded u_m. This implies that (3.14) is bounded-input bounded-output (BIBO) stable.

A challenging problem in the LMF approach is to choose the reference model appropriately. It not only must reflect the desired system behavior, but also must be reasonably chosen so that the plant can follow its trajectory closely. The Erzberger's condition gives indications on the constraints of the reference model for the PMF. It can be used to check whether the existing reference model satisfies the PMF conditions. However, as we can see in (3.11), it does not give much information on how to select (A_m, B_m, C_m). In the design process, what is needed is a guideline that can be used to select the reference model so that it will satisfy the Erzberger's condition. We shall look into frequency domain interpretation of the PMF conditions to gain additional insight to the problem.

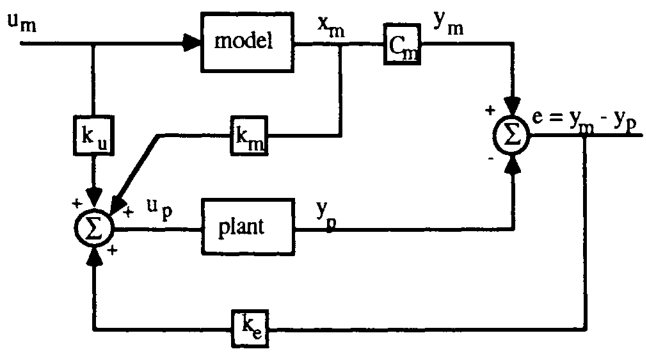


Figure 2 The control configuration of the explicit LMF.

3.2 Explicit LMF with Dynamic Compensators [13]

In the RCS, the ideal goal is to develop a control system that is able to accommodate a large class of different system impairments so that the reconfigured systems behave exactly as prespecified. The explicit LMF approach described above is an approach that makes the output of a plant follow that of a reference model to a certain extent using constant feedback and feedforward gains. The exact match only happens for a particular class of the reference models which have been chosen subject to some severe constraints. In the control reconfiguration to accommodate system failures, these constraints sometimes seem to be too restrictive. In this section we will investigate the use of dynamic compensators, instead of constant ones, to loosen the restrictions on the reference models.

From Figure 2, it is straight forward to derive the following

$$\begin{aligned} e &= y_m - y_p \\ &= (I + Pk_e)^{-1}[\, T_m - P[k_m(sI - A_m)^{-1}B_m + k_u]]u_m. \end{aligned} \tag{3.16}$$

Clearly, a necessary and sufficient condition for the PMF is that

$$[\, T_m - P[k_m(sI - A_m)^{-1}B_m + k_u] = 0. \tag{3.17}$$

Now it remains to solve (3.17) with respect to k_m and k_u. Note that there are many solutions of (3.17). A simple solution is

$$[k_m, k_u] = [0, P_{ri}T_m] \tag{3.18}$$

where P_{ri} is defined as the right inverse of P, i.e. $PP_{ri} = I$, assuming it exits. It was shown in [16] that the conditions for k_u to be proper and stable is that the reference model T_m is chosen such that it is 'more proper' than the plant and it has as its zeros all the RHP zeros of P together with their zero structures. It was also shown that the right-inverse of P can be calculated using a state-space algorithm which has good numerical properties. Note that the complexity of the compensator k_u is dependent on how different the reference model T_m is from the open-loop plant, P. This can be seen clearly from (3.18), where $k_u = P_{ri}T_m$. For example, if only a pair of open-loop poles are undesirable, that is, the poles and zeros of P and T_m are the same except one pair of poles, then k_u is a second order compensator since all the poles and zeros of P_{ri} and T_m are canceled except one pair of zeros of P_{ri} and one pair of poles of T_m. In case of failures, perhaps all open-loop zeros and poles will be shifted. However, only the unstable poles and dominant poles are of major concern in surviving the failures since their locations dominate how the system will behave in general. Therefore, in order to produce a fast and simple solution to keep the system running, T_m should be chosen close to the impaired plant P with exceptions of only a few critical poles.

The main advantage of this approach is that there are fewer restrictions on the reference model than before. The only restrictions are on the zeros of the reference model which are much more manageable than before. If the plant does not have RHP zeros, or its RHP zeros are unchanged after the failure, then the reference model is almost arbitrary except that it should be at least 'as proper' as P so that the compensator k_u is proper. The disadvantage of this approach is the increased complexity of the control system due to the higher order compensators required. This is a trade-off between the performance and complexity of the control system.

Note that there are many control configurations, other than that of Figure 2, that can be used for dynamic compensators. Here the same configuration is used for both constant and dynamic compensators because it is felt that the dynamic compensator can be used in conjunction with the constant compensators for the reconfigurable control purposes. As is mentioned earlier, there are two steps in the accommodation of failures. First, the impaired system must be stabilized. In the explicit LMF approach, this is accomplished via the implementation of the stabilizing gain k_e. This must be executed quickly to prevent catastrophic results from happening. Once the system is stabilized, it gives time to the control reconfiguration mechanism to manipulate the compensators to obtain better system performance. Assuming, by this time, that the model of the impaired system is available, a reference model should be chosen which has the desired behavior for the system under the specific system failure. Once the reference model is chosen, either the constant or the dynamic compensators can be computed and implemented as explained above. The choice of the types of the compensator depends on the performance requirements and the limitations on the complexity of the compensators.

3.3 An Alternative Explicit Linear Model Following Design [7,8]

Many control applications, especially the RCS, require that the structure of the system is simple, the design process is straightforward, and the computational complexity is kept minimum. The classical LMF approach meets these requirements quite well. Its main drawback is its inability to guarantee the performance of the closed-loop system. In this Section, an alternative linear model following approach is introduced for reconfigurable control design. This approach maintains the simplicity of the standard LMF methods and guarantees PMF under a very reasonable assumption.

The system configuration is shown in Figure 3. Here, the mathematical model of the nominal closed-loop system is chosen as the reference model while the plant is the impaired open-loop system. Let the reference model be given by equation (3.1) and the plant be represented by equation (3.2). The tracking error e is defined as the difference between the outputs of the plant and the reference model and hence allows the plant and the reference model to be of different dimensions. Clearly,

$$\dot{e} = C_m(A_m x_m + B_m u_m) - C_p(A_p x_p + B_p u_p)$$

From Figure 3, the control input u_p is of the form,

$$u_p = k_m x_m + k_u u_m + k_e e - k_p x_p \tag{3.19}$$

Therefore, $\dot{e}$ can be written as

$$\dot{e} = (-C_p B_p k_e)e + (C_m A_m - C_p B_p k_m)x_m + (C_m B_m - C_p B_p k_u)u_m + (C_p B_p k_p - C_p A_p)x_p \tag{3.20}$$

Note that PMF is achieved if $\{ k_m, k_u, k_p \}$ are chosen such that last three terms in (3.20) are zeros. That is, the sufficient conditions for PMF are

$$C_m A_m - C_p B_p k_m = 0 \tag{3.21}$$

$$C_m B_m - C_p B_p k_u = 0 \tag{3.22}$$

$$C_p B_p k_p - C_p A_p = 0 \tag{3.23}$$

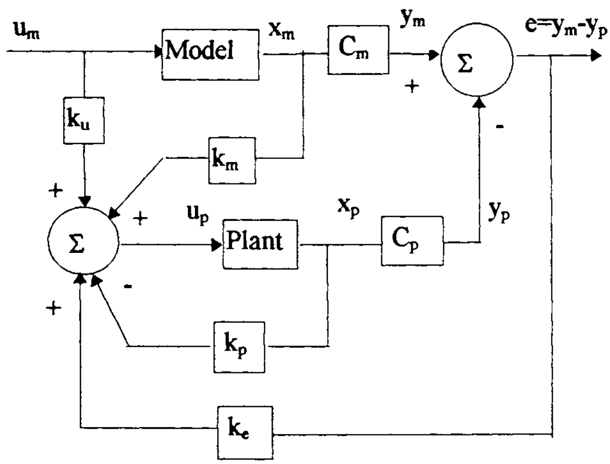


Figure 3. An alternative explicit model following design

The least square solutions of the above equations are given by

$$k_m = (C_pB_p)^+(C_mA_m) \tag{3.24}$$

$$k_u = (C_pB_p)^+(C_mB_m) \tag{3.25}$$

$$k_p = (C_pB_p)^+(C_pA_p) \tag{3.26}$$

where $(M)^+$ represents the pseudo inverse of the matrix M.

Substituting (3.24-3.26) in (3.20), the conditions for PMF can be rewritten as

$$[I - C_pB_p(C_pB_p)^+]C_mA_m = 0 \tag{3.27}$$

$$[I - C_pB_p(C_pB_p)^+]C_mB_m = 0 \tag{3.28}$$

$$[I - C_pB_p(C_pB_p)^+]C_pA_p = 0 \tag{3.29}$$

Note that these conditions are satisfied if C_pB_p has full row rank, or C_mA_m, C_mB_m, and C_pA_p are all in the null space of $[I-C_pB_p(C_pB_p)^+]$. From the above discussion, the following lemma is shown.

<u>**Lemma 3.1**</u>: PMF is achieved for the system in Figure 3 if $\{k_m, k_u, k_p\}$ are obtained from equations (3.24-3.26), and conditions in (3.27-3.29) are satisfied.

Remark: Compare conditions (3.27-3.29) to Erzberger's conditions (3.11), they are clearly more relaxed. Note that the conditions (3.27-3.29) are always satisfied when C_pB_p has full row rank, which is a reasonable assumption in control applications. Note that, in practice, the output and input variables are usually linearly independent, which means C_p and B_p matrices have full row and column ranks, respectively. Furthermore, it is well known that one must have at least as many inputs as outputs to manipulate each output independently. Therefore, C_pB_p matrix usually has at least as many columns as rows and a full row rank.

Lemma 3.2: The closed loop system in Figure 3 is asymptotically stable if and only if $(A_p\text{-}B_pk_p\text{-}B_pk_eC_p)$ has all its eigenvalues in the left half plane.

Proof: The closed-loop system can be represented by

$$\begin{bmatrix} \dot{x}_m \\ \dot{x}_p \end{bmatrix} = \begin{bmatrix} A_m & 0 \\ B_pk_m + B_pk_eC_m & A_p - B_pk_p - B_pk_eC_p \end{bmatrix} \begin{bmatrix} x_m \\ x_p \end{bmatrix} + \begin{bmatrix} B_m \\ B_pk_u \end{bmatrix} u_m$$

$$\begin{bmatrix} y_m \\ y_p \end{bmatrix} = \begin{bmatrix} C_m & 0 \\ 0 & C_p \end{bmatrix} \begin{bmatrix} x_m \\ x_p \end{bmatrix} \tag{3.30}$$

The closed loop system eigenvalues are those of the model A_m and those of $(A_p -B_pk_p-B_pk_eC_p)$. The reference model is always a stable system and A_m has all its eigenvalues in the left half plane. Hence the closed-loop system is asymptotically stable if and only if $(A_p -B_pk_p-B_pk_eC_p)$ has all its eigenvalues in the left half plane.

Note that the stability is the most important design criterion and must be maintained at all cost. If C_pB_p has full row rank and a stabilizing k_e can be found such that $(A_p-B_pk_p-B_pk_eC_p)$ is stable with k_p obtained from equation (3.26), then the closed-loop system is stable and PMF is achieved. However, when such k_e does not exist, a trade-off between tracking and stability must be made. In addition, the error dynamics should also be considered to ensure proper transient response. Based on these considerations, a design procedure is proposed as follows:

Design Procedure:

Step 1: Determine $\{k_m, k_u\}$ from equations (3.24) and (3.25).

Step 2: Find k_p and k_e such that the closed-loop system is stable, i.e. $(A_p-B_pk_p-B_pk_eC_p)$ has all its eigenvalues in the LHP, the error dynamics is acceptable, and $|C_pB_pk_p - C_pA_p|$ is minimized.

Step 3: Implement the new feedback and feedforward gains $\{k_m, k_u, k_p, k_e\}$.

Remark: In many applications, step 2 can be easily carried out by finding k_p from equation (3.26) and k_e as a stabilizing gain. Satisfactory results were obtained even when the conditions for PMF were not met. More details can be found in [7,8].

IV. OTHER TECHNIQUES

The RCS design problem has drawn the attention of many researchers and a wide range of methods have been published, including Linear-quadratic (LQ) control methodology [22,23,25,27,28,30], eigenstructure assignment [18,29,33], adaptive control [9, 31,37], H_∞ [40], knowledge-based systems [1,20,23], fuzzy logic [6,26], Neural Networks[5,41] , to name a few. These methods exhibit various degrees of success in simulations and are briefly discussed in this section.

Linear Quadratic Regulator Approach

In LQ design, the feedback gain is calculated by solving the Algebraic Riccati Equation (ARE) to minimize a quadratic performance index. The stability of the system is guaranteed provided that the impaired system is stabilizable. A direct application of LQ method in RCS, as shown in [27], is to solve the ARE with the model of the impaired plant and the new weights in the performance index. Another proposal [22] is to use a discrete form of LQ design in the context of proportional-integral implicit model-following framework.

The main advantage of LQ design is that the stability will be maintained and the computation is relatively simple. However, because the relation between the selection of weights and system performance is often ambiguous, the design usually requires several iterations before a satisfactory performance is achieved. Such characteristic seems to make it unsuitable for on-line implementation in RCS where controller redesign must be carried out automatically. To address such problem, Moerder et al [30] proposed to compute the LQ controllers off-line for anticipated failures and treat the RCS as gain scheduling problem. Obviously, unanticipated failures can not be accommodated in this manner.

Eigenstructure Assignment

Eigentructure assignment is a state feedback design method where both closed-loop eigenvalues and eigenvectors are assigned to achieve stability and performance. It is based on the concept that the solutions to eigenvalue assignment problem are not unique; by exploring the freedom in the selections of the state feedback gains, one can manipulate the closed-loop eigenvectors such that they are as close to the desired ones as possible. This usually leads to better transient response compared to pole-placement approach. Jin [29], Gavito et al [18] and Napolitano et al [33] applied this method to RCS design. It is interesting that both papers pointed out the high computation complexity of such algorithms. The problem of how to select appropriate eigenvalues and eigenvectors for the closed-loop system with an impaired plant is still also unresolved. For this method to be a viable candidate for RCS design, the algorithm must be further improved so that it can implementable in real-time without trial and error type of iterations.

Adaptive Control Scheme

Adaptive control is a well known design theory where controller is continuously updated to maintain the performance of the control system when there are significant dynamic variations in the plant. Several adaptive control methods were proposed for RCS [9,31,37], which have the benefits of integrating parameter estimation of the impaired system and control reconfiguration into one algorithm. Due to the nonlinear and time-varying nature of such systems, the stability and performance guarantee is hard to obtain even for relatively simple control problems. For reconfigurable control problems, it becomes much more challenging since the plants we deal with are generally complex with multiple inputs and outputs; furthermore, unlike most problems shown in the applications of adaptive control where plant parameters changes slowly, reconfigurable control problems usually involve sudden, significant changes in system dynamics which require immediate changes in control actions. It remains to be seen to what extent the adaptive control algorithms can keep up with the changes in system dynamics caused by various failures.

Intelligent Control Approach

The main avenues of Intelligent Control include expert systems, fuzzy logic, neural networks, all of which have been examined as possible candidates for RCS design. The Intelligent Control mechanism replaces the analytic controller in a conventional control system with a high level decision making scheme that generates the control signal based on a qualitative or heuristic understanding of the process. Typically the Intelligent Controller output is based on known trends or rules-of-thumb of how the process reacts in a dynamic environment. It is capable of capturing the expert knowledge and heuristics in various forms and incorporates this as a part of the control mechanism.

Yen [41] proposed a neural controller with self-learning capability to compensate slowly-varying as well as catastrophic changes in large space structures. Huang [23] uses a RCS framework that combines an expert system approach to fault analysis and a conventional model following approach to control reconfiguration. Chandler et al [5] extended their work based on the PIM and explored new controllers using Hopfield networks to identify system parameters and redesign control laws. Kwong et al [26] experimented with a Fuzzy logic based learning system for control reconfiguration.

V. Concluding Remarks

Reconfigurable control is a relatively new concept that has been explored by many researchers over the last ten to fifteen years. The account here is by no means complete. Instead, only a few methods that shown potential for practical implementation are discussed in details; a few interesting but yet fully developed methodologies are also introduced.

Reconfigurable control is a complex problem that involves failure detection and identification, system identification, control law reconfiguration and, above all, a high level decision making mechanism that decide when and how to redesign the controller in the face of system failures. Here, only the control reconfigurations techniques are discussed. Although much progress have been made, this is still a largely open research area because the existing control theory hardly addresses the need of real-time controller redesign. One must be careful when applying general design techniques, such as H_∞, to RCS because it is simply impossible to 'play with' design parameters such as weights in a trial and error manner in failure accommodation. What makes the reconfigurable control problem unique and challenging is that the control law redesign must be carried out in real-time without human intervention.

REFERENCES

1. Blokland W. and Sztipanovits J., 1988, Knowledge-Based Approach to Reconfigurable Control Systems, *Proceedings of the 1988 American Control Conference*, 1623-1628, June, 1988.
2. Bodson M., 1993, Identification with modeling uncertainty and reconfigurable control, *Proceedings of the 32nd IEEE Conference on Decision and Control*, Vol 3, 2242-7, Dec. 1993.
3. Caglayan A.K., Allen S.M., and Wehmuller K., 1988, Evaluation of a Second Generation Reconfiguration Strategy for Aircraft Flight Control Systems Subjected to Actuator Failure/Surface Damage, *Proc. of the IEEE 1988 National Aerospace and Electronics Conference*, 520-529.
4. Chandler P.R., 1984, Self-Repairing Flight Control System Reliability & Maintainability Program Executive Overview, *Proc. of the IEEE 1984 National Aerospace and Electronics Conference*, 586-590.
5. Chandler P., Mears M., and Pachter M., On-line optimizing networks for reconfigurable control, *Proceedings of the 32nd IEEE Conference on Decision and Control*, Vol 3, 2272-2277, Dec. 1993.

6. Copeland R.P. and Rattan K.S, A fuzzy logic supervisor for reconfigurable flight control systems, *Proceedings of the IEEE 1994 National Aerospace and Electronics Conference NAECON 1994* . Vol.1, 579-86, 1994.

7. Dhayagude N. and Gao Z., A Novel Approach to Reconfigurable Control Systems Design, to appear in *Proc. of 1995 AIAA Guidance and Control Conference*, August 1995.

8. Dhayagude N., Linear Model Following Design Approach and Stability Robustness in Reconfigurable Control Systems, M.S. Thesis, Department of Electrical Engineering, Cleveland State University, April 1993.

9. Dittmar C.J., 1988, A Hyperstable Model-Following Flight Control System Used for Reconfiguration Following Aircraft Impairment, *Proc. of American Control Conference*, 2219-2224, June 1988.

10. Erzberger H., 1968, Analysis and Design of Model Following Control Systems by State-Space Techniques, *Proc. of JACC*, pp.572-581, 1968.

11. Eslinger R.A., and Chandler P.R., 1988, Self-Repairing Flight Control System Program Overview, *Proc. of the IEEE 1988 National Aerospace and Electronics Conference*, 504-511.

12. Gao Z., and Antsaklis P.J., 1991, On the Stability of Pseudo-Inverse Method for Reconfigurable Control Systems, *International Journal of Control*, VOL. 53, No. 3, 717-729.

13. Gao Z., and Antsaklis P.J., 1993, Reconfigurable Control System Design via Perfect Model Following, *International Journal of Control*, VOL. 56, No. 4, 783-798.

14. Gao Z., 1990, Reconfigurable Control: Analysis and Design, Ph.D. Dissertation, Department of Electrical Engineering, University of Notre Dame, August, 1990.

15. Gao Z., and Antsaklis P.J., 1992, Actuator and Sensor Failure Accommodation with Incomplete Information, *Proc. of the First IEEE Conference on Control Applications*, pp.890-892.

16. Gao Z., and Antsaklis P.J., 1989, On Stable Solutions of the One- and Two-Sided Model Matching Problems, *IEEE Transactions on Automatic Control*, Vol. 34, No. 9, pp. 978 - 982.

17. Gao Z., and Antsaklis P.J., 1989, Explicit Asymmetric Bounds for Robust Stability of Continuous and Discrete-Time Systems, *IEEE Transactions on Automatic Control*, Vol 38, No. 2, pp. 332-335, February, 1993.

18. Gavito V.F., and Collins D.J., 1987. Application of Eigenstructure Assignment to Self Reconfiguring Aircraft MIMO Controllers, *Proc. of 1987 AIAA Guidance and Control Conference*, 1-12.

19. Golub G.H., and Van Loan C.F., 1983, *Matrix Computations*, (Baltimore: The John Hopkins University Press).

20. Handelman D.A. and Stengel R.F., 1989, Combining Expert System and Analytic Redundancy Concepts for Fault Tolerance Flight Control, *J. of Guidance, Control, and Dynamics*, Vol.12, 39-45.

21. Howell W.E., Bundick, W.T., Hueschen, R.M. and Ostroff, A.J., 1983, Restructurable Controls for Aircraft, *Proc. of 1983 AIAA Guidance and Control Conference*, 646-651.

22. Huang C.Y. and Stengel R.F., 1990, Restructurable control using proportional-integral implicit model following. *Journal of Guidance, Dynamics & Control*, Vol.13, No.2, 303-339.

23. Huang C.Y., 1989, A Methodology for Knowledge-Based Restructuralbe Control to Accommodate System Failures, Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, Princeton University, August, 1989.

24. Huber R.R., and McCulloch B., 1984, Self-Repairing Flight Control System, Society of Automotive Engineers Technical Paper Series, 841552, Aerospace Congress & Exposition, Long Beach, California.

25. Joshi S.M., 1987, Design of Failure - Accommodating Multiloop LQG - Type Controllers, *IEEE Transactions on Automatic Control*, AC-32, 740-741.

26. Kwong, W.A., K.M. Passino, and S. Yurkovich, 1994, Fuzzy Learning Systems fro Aircraft Control Law Reconfiguration, *Proceedings of 1994 IEEE International Symposium on Intelligent Control*, 333-338.

27. Looze D.P., Weiss, J.L., Eterno, J.L., and Barrett, N.M., 1985, An Automatic Redesign Approach for Restructurable Control Systems, *Control Systems Magazine*.

28. McLean D., and Aslam-Mir S., Optimal integral control of trim in a reconfigurable flight control system, *Control Engineering Practice*, vol.2, no.3, 453-60, June 1994.

29. Jin J., Design of reconfigurable control systems using eigenstructure assignments, *International Journal of Control*, Vol. 59, No. 2395-410.

30. Moerder D.D., Halyo, N.; Broussard J.R., and Caglayan A.K., 1989, Application of precomputed control laws in a reconfigurable aircraft flight control system, *Journal of Guidance, Control, and Dynamics*, Vol.12, no.3, 325-33

31. Morse W.D. and Ossman K.A., 1990, Model following reconfigurable flight control system for the AFTI/F-16, *Journal of Guidance, Dynamics & Control*, Vol.13, No.6, 969-976.

32. Napolitano M.R. and Swaim R.L., 1991, New Technique for Aircraft Flight Control Reconfiguration, *Journal of Guidance, Dynamics & Control*, Vol.14, No.1, 184-190.

33. Napolitano M.R. and Swaim R.L., 1991, Redesign of the feedback structure following a battle damage and/or a failure on a control surface by

eigenstructure assignment, *Proc. of 1991 AIAA Guidance and Control Conference*, 247-254.

34. NTSB, 1979, National Transportation Safety Board Accident Report of the American Airlines DC10 Crash at Chicago -- O'Hare International Airport, May 25, 1979, NTAB-AAR-79-17.

35. Ochi Y., and Kanai K., 1991Design of Restructurable Flight Control Systems Using Feedback Linearization, , *Journal of Guidance, Dynamics & Control,* Vol.14, No.5, 903-909, Sept. 1991.

36. Ostroff A.J, 1985, Techniques for Accommodating Control Effector Failures on a Mildly Statically Unstable Airplane, *Proc. of American Control Conference*, 906-903.

37. Pogoda D.L. and Maybeck, P.S. 1989, Reconfigurable flight controller for the STOL F-15 with sensor/actuator failures, *Proc. of the IEEE 1989 National Aerospace and Electronics Conference*, 318-324.

38. Rattan K.S., 1985, Evaluation of Control Mixer Concept for Reconfiguration of Flight Control System, *Proc. of the IEEE 1985 National Aerospace and Electronics Conference*, 560-569.

39. Raza S.J., and Silverthorn J.T, Use of Pseudo - Inverse for Design of a Reconfigurable Flight Control System, *Proc. of 1985 AIAA Guidance and Control Conference*, 349-356.

40. Wu N. E., 1993, Reconfigurable Control Design: Achieving Stability Robustness and Failure Tracking, *Proceedings of the 32nd Conference on Decision and Control*, 2278-2283, December 1993.

41. Yen G.G., 1994 Reconfigurable Learning Control in Large Space Structures, *IEEE Transactions on Control Systems Technology*, Vol. 2, No. 4, 362-370, December, 1994.

42. Zhou Z and Khargonekar P.P, 1987, Stability Robustness Bounds for Linear State-Space Models with Structured Uncertainty, *IEEE Transactions on Automatic Control,* AC-32, pp.621-623.

This Page Intentionally Left Blank

Techniques in Deadbeat and One-step-ahead Control

Tom T. Hartley
Robert J. Veillette

Department of Electrical Engineering
The University of Akron
Akron, OH 44325-3904

Gerald Cook

Department of Electrical and Computer Engineering
George Mason University
Fairfax, VA 22030

I. INTRODUCTION

Discrete-time dynamic systems present some interesting possibilities that systems described in continuous-time do not. For example, in a noise-free environment, it is possible to measure the system inputs and outputs and algebraically determine the previously unknown system parameters. In a noisy environment, the procedure can be extended by closed-form optimization using one of the many least squares techniques. This procedure is referred to as system identification. If the discrete-time system parameters are known, it is possible to algebraically cancel the polynomials representing the system dynamics via feedback compensation to obtain a finite settling time with zero error for a given reference input. This strategy is referred to as deadbeat control. Another control technique that can be used when the system parameters are known involves algebraically subtracting out the system dynamics via feedback compensation. Doing this allows not only finite settling time, but also tracking of any input signal with a one step delay between the input and the output. This strategy is referred to as one-step-ahead control.

System identification, deadbeat control, and one-step-ahead control are possible because discrete-time systems are described by difference equations, which define an algebraic relation between the system inputs and outputs. Such procedures cannot be applied to continuous-time systems, whose inputs and outputs are related by differential equations. As a result, for example, deadbeat response in a continuous-time system cannot be achieved via linear time-invariant feedback. No matter how rapid the step response of a continuous-time control system is made, the exponentially decaying error vanishes only as time goes to infinity.

This chapter explores the development of deadbeat control and one-step-ahead control. System identification has been dealt with extensively elsewhere, and will not be discussed here, except as it pertains to the application of the two control techniques. Although deadbeat control and one-step-ahead control each have a long history, the two were developed in separate research streams, and it was many years before these streams merged. Furthermore, there seems to be some general misunderstanding about the specific meaning of deadbeat control. This discussion is intended to compare and contrast deadbeat and one-step-ahead control strategies, to contribute to the unification of the two areas, and to clear up some possible points of confusion.

The discussion is organized as follows. Deadbeat control is considered first. This is divided into sections describing the standard deadbeat control derivations, the origins of deadbeat control, and modern developments. One-step-ahead control is then considered. It will also be divided into sections on origins and modern developments. Some general observations will then be made. At the end is a chronological list of the work cited, as well as a list of related references. We have made every effort to make the bibliography accurate and complete. We offer our apologies to anyone whose work may have been misrepresented or omitted. Citations in the text are denoted by the author's name and the date.

II. DEADBEAT CONTROL

A. INTRODUCTION TO DEADBEAT DESIGN

1. Standard I/O Derivation

The simplest and best known derivation for deadbeat control begins with the unity-feedback closed-loop control system shown in Figure 1. The closed-loop transfer function is

$$T(z) = \frac{C(z)}{R(z)} = \frac{G(z)H(z)}{1+G(z)H(z)}\,. \tag{1}$$

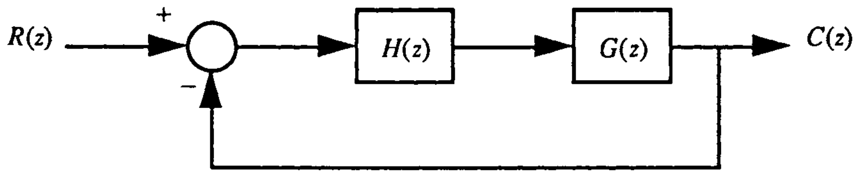


Figure 1. Discrete-time unity-feedback control system.

Given the desired transfer function $T(z)$, it is a simple task to solve for the required compensator $H(z)$. Performing the algebra, the result is

$$H(z) = \frac{1}{G(z)} \cdot \frac{T(z)}{1-T(z)} \, . \tag{2}$$

In general, this controller may require the complete cancellation of the plant $G(z)$. Therefore, to avoid unstable pole-zero cancellations, it must be assumed that the plant is stable and minimum phase. To achieve minimum settling time in response to a step input, a desirable closed-loop transfer function would be $T(z)=1/z$. The resulting deadbeat controller is

$$H(z) = \frac{1}{G(z)} \cdot \frac{1/z}{1-1/z} \tag{3}$$

or

$$H(z) = \frac{1}{G(z)} \cdot \frac{1}{z-1} \, . \tag{4}$$

If $G(z)$ has relative degree no greater than one, then this controller is realizable, and the forward path transfer function is simply a discrete-time integrator. If $G(z)$ has relative degree $m>1$, then the controller (4) would be improper. In this case, the fastest achievable response is represented by the closed-loop transfer function $T(z)=1/z^m$, which is obtained by choosing the proper controller

$$H(z) = \frac{1}{G(z)} \cdot \frac{1}{z^m-1} \, . \tag{5}$$

The root locus plots for the cases $m=1,2,3$ are shown in Figure 2. In each case, the forward-path transfer function includes a discrete-time integrator, which ensures zero steady-state error for step inputs. The choice of unity gain causes

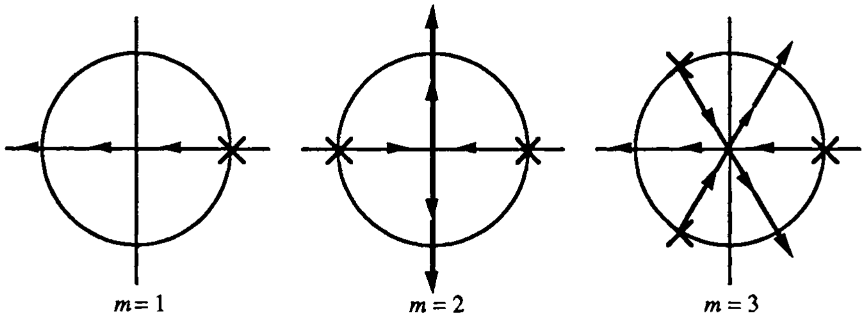


Figure 2. Root locus plots for deadbeat control systems.

all the closed-loop poles to coincide at z=0, so that the steady-state condition is reached in a finite number of timesteps.

By a more general choice of the closed-loop transfer function $T(z)$, it is possible to obtain finite settling time responses for other classes of reference inputs. The desired class of inputs must be specified *a priori*. It is not possible to design a controller that will give finite settling time responses for an unspecified class of references.

It should be noted that other design techniques for finite settling time are available. For example, the input-output pole placement technique (Astrom and Wittenmark, 1984) allows the placement of $2n$ closed-loop poles at z=0 with no pole-zero cancellations.

2. State-variable derivation

The fundamental idea behind deadbeat control in a state-variable setting is the placement of all of the closed-loop system poles at z=0. Such pole placement allows finite settling to a step input in n timesteps. Let the plant be given by a controllable state-variable description

$$x(k+1) = Ax(k) + Bu(k), \tag{6}$$

and assume the state variables are available for feedback. A state-feedback control law has the form

$$u(k) = -Kx(k) \tag{7}$$

which yields

$$x(k+1) = (A-BK)x(k) \,. \tag{8}$$

Then the deadbeat controller design problem reduces to determining the feedback gain matrix K such that all the eigenvalues of $A-BK$ lie at $z=0$. Although standard pole placement techniques, such as Bass-Gura, can be used, many variations are possible in the choice of K, particularly in the multi-input case.

B. IN THE BEGINNING ... (SISO)

Deadbeat control traces its beginnings almost back to the beginning of sampled-data systems. The first major comprehensive collections of control techniques became available publicly near the end of World War II. The two major works, both released in 1945, were those of Bode (1945) and MacColl (1945). The work of Bode is now recognized as a timeless classic. It is grounded solidly in mathematics and full of interesting relationships involving frequency response characteristics and stability. By contrast, MacColl's book seems to be little known, although it is really more of a control book than Bode's. It contains one of the first discussions of "sampling servomechanisms." MacColl states that much of his discussion follows from that of C.E. Shannon, and that "the first treatment along these lines" was due to G.R. Stibitz. The discussion contains frequency response stability information but no discussion of any discrete-time transform theory.

The first major theoretical improvement in the discrete-time systems area was made by Hurewicz (1947), who considers systems with "pulsed data." His insights apparently come "out of the blue," as his only reference is to Titchmarsh's *Theory of Functions* (Oxford, 1932). His major contribution is the use of the variable z to represent e^{sT}. Using this, he is able to develop the familiar z-domain stability properties of discrete-time systems. He also discusses the use of "clamping devices," which we would recognize as zero-order holds. Given the widespread use of the techniques first introduced there, (Hurewicz, 1947) is definitely a remarkable and important work in the history of control systems. Additional references to the development of discrete-time system theory can be found in the chronological bibliography of (Tou, 1959). The present discussion will now turn directly toward deadbeat control.

The first significant paper in the theory of deadbeat control appears to be that of Porter and Stoneman (1950). It should be noted that Porter and Stoneman refer to sampled data systems as "pulse-monitored servo systems" and credit Hazen ("Theory of Servomechanisms," *J. Franklin Inst.*, vol. 218, pp. 279, 1934) with the idea, which he called "definite correction" servomechanisms. In retrospect, this paper contains a peculiar treatment of discrete-time systems.

It contains continuous-time integrators (which at the time came in a variety of interesting mechanical and electrical forms) driven by a sampled signal, with the purpose of reconstructing the data between sampling instants. After showing that it is possible to reconstruct the signal with relatively small error, they mention possible applications, including tracking. The Discussors for this paper were Holt Smith, Ghalib, Barker, and Wallis, and an appendix by Lawden is included. In his discussion, Barker presents what would appear to be the first representation of a deadbeat step response using Porter and Stoneman's data reconstructors in Figures A and B. Clearly it was a simpler time, as in his reply to the Discussors, Major Stoneman states that Dr. Porter was unable to add his comments, as he was "in Canada." Today it would have been all too easy to disturb the absent Dr. Porter using any of several media.

The next major contribution is by Lawden (1951). He uses the z-transform theory of Hurewicz (1947) to clarify the analysis of Porter and Stoneman (1950). There he says that Porter and Stoneman (1950) have used the characteristic equation $z^m=0$, although this is difficult to see in their paper. Lawden then states that "such a system is over-responsive to errors" and introduces the staleness factor a such that the closed loop characteristic equation is $(z-a)^m=0$, where $|a|<1$ for stability. An example of this approach is also given. It is interesting to note that the Evans root locus technique had just been introduced and was not at all well known at this time, yet these researchers were already using the closed-loop pole placement idea.

The next year, 1952, brought three important contributions. One is that of Holt Smith *et al.* (1952). This paper builds upon the work of Porter and Stoneman (1950) and Lawden (1951). The Porter and Stoneman approach is further developed here, and the staleness factor a is named. Extensive examples are presented. R.H. Barker was again a Discussor for this paper. In his discussion, he "presents a slightly more physical approach to the sequence transform method." A major point of his discussion is the output between samples, where he appears to have introduced the modified z-transform.

The second major contribution of 1952 was that of Barker (1952) himself. There he clearly describes the "sequence transform method" for the development of "pulse transfer functions" when dealing with "sampling servo systems." It is basically an extension of (Hurewicz, 1947), but it is clear, concise, and complete. Apparently it contains the first organized table of z-transforms, including modified z-transforms. The transform pairs are given alongside their Laplace domain counterparts, making the table particularly useful. The paper gives examples demonstrating deadbeat response. It also introduces the idea of general prototype systems: "A possible approach is to synthesize a system having a characteristic equation chosen beforehand and known to possess a certain degree of stability. The particular case where all the roots are real and equal has been studied by Holt Smith, Lawden, and Bailey." Clearly this is another important paper in the development of discrete-time control systems.

The third major contribution was the classic paper of Ragazzini and Zadeh (1952). This paper solidified many components in the theory of sampled-data systems. Basically the purpose of the paper was (a) to unify the results of Hurewicz (1947), MacColl (1945), and Linville (1951), and (b) to determine input-output behavior for sampled-data systems in various configurations. This appears to contain the first use of the terminology "z-transform." A very short table of z-transform pairs is given as well as a table of closed-loop transfer functions with samplers at various places. The paper contains a clear discussion of hold devices, and demonstrates the zero-order-hold method. The Discussors are J.M. Salzer, B.M. Brown, and W.K. Linville. Salzer states that "the paper fills a clear need in this underpublished field, and it presents the topic concisely and illuminatingly." Salzer also suggests that the definition $z{=}e^{-sT}$ would be much preferable than the given $z{=}e^{+sT}$. Ragazzini and Zadeh respond that they have only followed (Hurewicz, 1947) and that they agree with Salzer. In his usual fashion, Brown expounds on the calculus of finite differences and shows that $E{=}1{+}\delta{=}e^{TD}$, where E is the delay operator, δ is the difference operator, and D is the differential operator. He then introduces a time-domain sampling operator S. Ragazzini and Zadeh respond that the calculus of finite differences is to the z-transform theory as the Heaviside operational calculus is to the Laplace transform. We will not enter into this argument, but point out that many of the important and useful contributions of (a) Heaviside, Bush, and Carson, in continuous-time; and (b) Boole, Jordan, Steffensen, Fort, and Milne-Thompson in discrete-time are all but unknown to most modern control engineers and system theorists. In any case, the important paper by Ragazzini and Zadeh (1952) is often considered as the original reference in sampled-data control theory.

Two years later came the next major contribution, that of Bergen and Ragazzini (1954). This paper laid the groundwork for all future work in the area of finite-settling-time (deadbeat) control, and explains many of the important concepts involved. As such, it is probably the primary reference in the area, although many of the important ideas were established in the work of others as discussed above. Clearly, as Porter and Stoneman are referenced first by Bergen and Ragazzini, they should receive the credit for introducing the deadbeat idea. The other authors, Linville, Lawden, Holt Smith, Bailey, and particularly Barker should also be remembered for their continued development of the concept. However, in this paper, Bergen and Ragazzini very clearly present the idea of finite-settling-time control systems and should be credited for clarifying and grounding the area. Many concepts are named for apparently the first time. For example, the older idea of choosing feedback to obtain a particular closed-loop transfer function is discussed, and the chosen closed-loop transfer function is called a "prototype form." Equations are given for designing the compensator for a particular prototype form. Furthermore, stability limitations on the open-loop plant are given to achieve these prototype forms. "Finite settling time" is then discussed for a given input "test signal." Equations are then given, along with a table, for obtaining finite settling time for a given reference. Then "minimum settling time" is discussed which leads to "minimum prototype forms." Limitations are discussed including limited control input and uncertain parameters. Example implementations are included.

The Discussors for this paper are J.M. Salzer, R.E. Kalman, and R.H. Barker. The first Discussor, Salzer, states near the beginning of his comments that "One advantage of a sampled-data compensator over a continuous-data compensator stressed in the paper is that a deadbeat response to a step or ramp input can be achieved. The expression 'deadbeat' means that after a fixed and finite time interval the response is exactly as the input. What needs additional emphasis, although implicit throughout the paper, is that the response will match the input only at the sampling instants. If we should consider the response between samples, we would find that its beat is indeed not dead at all." This would appear to be the first usage of the term "deadbeat" in the literature and we thus credit J.M. Salzer for this.

Although Salzer's use of the term "deadbeat" leaves little room for confusion, a certain degree of uncertainty has arisen over the years concerning the meaning and origin of the expression. English dictionaries list the following definitions:

> 1. Stopping without oscillation (from: DEAD + BEAT (oscillation)). 2. Lacking recoil. 3. A person who does not pay his debts. 4. A lazy or lethargic person, loafer (from: DEAD (completely) + BEAT (exhausted)). 5. In music, a beat where no one plays (from: DEAD (quiet) + BEAT (fixed time interval)).

Although Salzer clearly had definition (1.) in mind, a student hearing the term for the first time may initially think of definitions (3.) or (4.). It should be noted that neither Salzer nor any of the consulted dictionaries hyphenate the word deadbeat, as have many unknowing control engineers since. An alternative usage of the term deadbeat, occasionally heard in financial circles, is almost the opposite of definition (3.) It refers to any person who regularly pays off all his credit-card debt in full each month. It is not certain whether the term is meant to be uncomplimentary, since such a customer generates less income to the creditor, or whether it refers to the regularity of the payments. One other definition was found in the control literature. Fahmy and O'Reilly (1983) state that "Such controllers are called 'dead-beat' (DB) controllers because 'they beat the system to a dead stop'." (Although they appear to be quoting someone, they give no reference.) They use the hyphen in deadbeat, and their definition seems far from the intended, although a reasonable rationalization.

Another problem with the meaning of the term deadbeat, is that it has come to primarily imply minimum settling time to a step input, while it was initially intended to mean minimum settling time to any input. To underscore this, some modern texts will pose problems such as: Design a deadbeat controller for the following system. Of course, deadbeat to a step input is intended.

The second Discussor for (Bergen and Ragazzini, 1954) is R.E. Kalman. He makes several positive statements about the importance of the "time-domain approach" used by Bergen and Ragazzini, saying that "this constitutes a much broader viewpoint than the frequency-response representation ..., and frequently gives deeper insight into the problem." Kalman then goes on to describe what appears to be the first discussion of posicast control, although he does not call it that. (It is discussed more fully by O.J.M. Smith (1958), who states that "The control motion is like casting a fly; hence the name 'positive-cast,' or Posicast, control.")

The third Discussor for the paper is R.H. Barker. He correctly states that much of what is in the paper was presented in his own earlier work. Had he stopped there, history might have treated him more kindly. Instead, he proceeded to attack the paper on several points. He argues about "pulse" versus "pulsed" transfer functions, and at one point degenerates into "The sentence following equation 6 is loosely worded..." Although it is impossible to ascertain his true feelings at the time, it would appear that he was none too happy about the publication of this paper. In their response, Bergen and Ragazzini politely state "We have long been acquainted with the outstanding work done by Mr. Barker and his British colleagues in the field of sampled-data systems. Reference 6 [Barker (1952)] is a classic in the field and should be read by all who are interested in learning the fundamentals of sampled-data systems." They then proceed to disagree with much of Barker's argumentation. It would seem that a great amount of insight has been lost by not including the comments of Discussors in many modern journals. Much of the important material of the papers considered in this survey was found in the discussions and the replies.

The year 1956 brought two important contributions. Jury (1956) contributed another classic in the development of sampled-data control systems. Modified z-transforms are used extensively and credit for their development is given to Barker. Tables are included. This paper presents extensive closed-loop stability analysis, and discusses the discrete-time root locus, but does not mention deadbeat. The other 1956 paper is (Bertram, 1956). Here Bertram discusses finite-settling-time control and suggests that, in addition to shifting the denominator from z^n to $(z-a)^n$, a numerator polynomial should be added to allow a compromise between the responses to various inputs, rather than just the one for which the deadbeat controller was developed. The Discussors are E.I. Jury and W. Schroeder together. They state that "The main contribution of this paper is its indication that if a system is to follow both step and ramp inputs, a compromise between settling time and overshoot at the sampling time must be made." They assert that "only the poles of the plant may be cancelled" and "any attempt to cancel a zero only results in the addition of an undesirable pole" and "can only result in a large settling time for the continuous response." They also stress the importance of considering the intersample behavior.

Apparently motivated by their Discussorship, the next year Jury and Schroeder (1957) expand upon their interest in the intersample behavior of deadbeat control. They present a clear and organized procedure for eliminating the intersample ripple, thus providing true finite settling time, for the sampled-data system both at the sampling instants and between the sampling instants.

The year 1958 was a good year for the coalescing of material, as the first two major works on sampled-data control systems were released. These were (Jury, 1958) and (Ragazzini and Franklin, 1958). Both works contain large sections on sampling, hold devices, compensation techniques, intersample behavior, modified z-transforms, and, of course, deadbeat control. It is clear that deadbeat control was one of the important concepts in the area, even with its limitations. Also that year, the excellent book by O.J.M. Smith on continuous-time systems (1958) was released. Here the famed Smith predictor is presented and discussed at length for various system types, as well as posicast control. Some very interesting problems can be found at the end of the Chapters, particularly the Chapter on time delays. It should be noted that both the Smith Predictor and Posicast ideas are directly related to settling time. The Smith predictor has the effect of shifting the plant delays outside the feedback loop, permitting more conventional control design methods to be used. Posicast control, an idea that is still found useful today (Cook 1986; Singer and Seering, 1990), consists of a feedforward compensator that achieves a finite settling time by delaying a portion of the input signal.

In 1959, Schmidt (1959) presents a paper that correctly demonstrates that finite-settling-time designs are very sensitive to uncertainty. He states "that since the performance of such sampled data systems cannot be achieved in practice they should be avoided." He states that "designs using pole cancellation should be avoided for the following reasons: ... the abrupt motion of the digital controller ... may lead to saturation of the input to the plant" and "if undesirable poles are cancelled ... with respect to the command input, they will not be cancelled with respect to the disturbance inputs, and an undesirable load disturbance response may result." The Discussor for the paper is Gene Franklin. He states "the primary purpose of this discussion is to drive an analytical nail in the coffin of the finite settling time design ..." and "despite the calculations that follow, the finite settling time design will ... probably remain a practical design for low order systems which must operate with very long sampling periods." The calculations that followed had the result that "the finite settling time design which places all poles at the origin of the z-plane is operating at a point of infinite sensitivity of root position to parameter change! A design which shifts the multiple roots from the origin to a point $z=a$ does not change the sensitivity picture." Furthermore "the larger the number of poles placed at a single operating point, the more sensitive will be the root locations to parameter changes." The author's reply is that Dr. Franklin's result "gives further proof as to why [finite-settling-time designs] should be avoided." Schmidt then makes the prophetic statement "further research would be desirable to determine ... how to design control systems which have only a small sensitivity with parameter variations." Despite these gloomy observations, the finite-settling-time, or deadbeat, idea has remained an important research area which continues to attract many paper writers.

Also in 1959, another great book on sampled-data theory was released by Tou (1959). Of course a large section on finite-settling-time compensators is included. This is an important reference, not only for its general contents, but also for the extensive year-by-year bibliography in the area of sampled-data systems up until 1959. Many references are given, including applications.

Finally, the papers of Kalman and Bertram (1958, 1959) were published, leading to the very important paper by Kalman (1960). This paper formalized many important ideas, including state-space representations, time-optimal control, and deadbeat control in state space, and introduced the concepts of controllability and duality. Here the deadbeat idea was to drive the system state to the origin in the minimum number of time samples. Kalman states that controllability "is the 'natural' generalization of the so-called 'dead-beat' control scheme...." Interestingly, he credits Oldenbourg and Sartorius (1951) with the discovery of deadbeat control, and Tsypkin (1950) and himself, as a Discussor in (Bergen and Ragazzini, 1954), with its later independent rederivation. These papers may be another origin of the deadbeat control idea; however, they do not appear in the mainstream of the historical evolution, but rather enter through Kalman. Many important contributions of Kalman to deadbeat control are discussed further by O'Reilly (1981).

Except for the paper by O.J.M. Smith (1963), the deadbeat control area was fairly quiet after this until the mid 1960's. This paper gave a simplified algorithm to solve the finite settling time problem, while still eliminating intersample ripple. Another paper is presented by Hyde and Li (1965) where the minimum settling time problem is solved with an assured stable compensator. In the same year, the book by Lindorff (1965) is published which contains the usual transfer-function finite-settling-time discussion, but which also contains a treatment of finite settling time in state space, perhaps the first in a text.

B. THE MODERN BAROQUE PERIOD

The widespread acceptance of the state space formulation for control system design initiated by Kalman (1960) provided a host of new topics for researchers in finite-settling-time control systems. Beginning in the late 1960's, many different techniques for multivariable finite-settling-time control have been developed. Our attempt to organize the various techniques into logical categories is presented below.

1. Geometric design

As mentioned earlier, the concept of controllability was originally identified with the possibility of deadbeat control in a state-space setting (Kalman, 1960). Kalman (1960) defines the deadbeat control problem generically as that of finding a control sequence such that the state of the single-input system

$$x(k+1) = A[x(k) + bu(k)] \tag{9}$$

at any time k would be transferred from $x(k)$ to the origin in the minimum possible number of steps. The solvability of this problem depends on the familiar rank condition on a controllability matrix, which leads in turn to a linear state-feedback solution

$$u(k) = Fx(k). \tag{10}$$

The later work extending the results to the multi-input case

$$x(k+1) = Ax(k) + Bu(k) \tag{11}$$

is typified by (Leden, 1977) and (Akashi and Imai, 1978). Leden (1977) considers deadbeat response in a particular output, but allows state feedback; Akashi and Imai (1978) also present observer-based solutions. Since these papers treat the multi-input case, there is additional design freedom relative to the single-input case.

The geometric method has also been used to address the problem of deadbeat state observation. For example, Adachi and Akashi (1982) give a procedure for constructing a minimum-order observer that reconstructs a linear function of the state variables in minimum time. They note that in some cases the minimum-time (less than n steps) observer may need to have higher order than a generic minimum-order observer that provides deadbeat response in n steps.

As pointed out by O'Reilly (1981), an element common to all the deadbeat control papers taking the geometric approach is that the development does not assume a linear feedback solution *a priori*, but hinges on properties of the vector spaces (range and null spaces, and invariant subspaces) associated with various system operators. Of course, all the solutions, just as that in the original work of Kalman, share the property that all the eigenvalues of the closed-loop system matrix $A+BF$ lie at the origin of the complex plane.

2. Pole-placement design

The general approach to deadbeat control problems via the geometric method seems to lead inexorably to state-feedback solutions; consequently, some authors following this line of research have given in on this point, and have assumed the state-feedback form of control as a starting point. Then, the deadbeat control design boils down simply to the choice of F for the placement of all the eigenvalues of $A+BF$ at the origin. The only design issue is the use of the freedom available for this pole placement in the case of a multi-input system (Fahmy and O'Reilly, 1983) or a singular A-matrix (Erol and Loparo, 1983). This design freedom is used to shape the finite duration transient response (i.e., to influence the zeros) of the closed-loop system.

The concept of deadbeat state-feedback design leads naturally, via the principle of duality, to that of deadbeat observer design for use in output-feedback implementations. Porter and Bradshaw (1975) present a simple argument, based on the Brunovsky canonical form, that any nth-order controllable and observable system can be driven from any initial state to the origin of the state space in at most $2n$ steps by use of a deadbeat state-feedback control design in conjunction with a deadbeat observer. The same authors (Bradshaw and Porter, 1976a,b) also investigate the use of deadbeat observers for multivariable tracking problems, and demonstrate that, aside from a finite-duration transient, perfect tracking is achieved for a class of polynomial inputs by use of a deadbeat state-feedback-plus-integral control design in conjunction with a deadbeat observer. O'Reilly (1980) develops the case of minimal-order deadbeat observers, including convenient expressions for the feedback and observer gains.

An alternative to state-feedback or observer-based design is the possibility of deadbeat control via constant output feedback (Seraji, 1975). This approach consists of the search for a constant matrix K for which all the eigenvalues of the matrix $A+BKC$ lie at the origin. Such a feedback matrix does not always exist.

3. Factorization approaches

Independent of the development of deadbeat control in a state-variable setting, a few new lines of work developed an input-output approach. Prominent in this arena was the research on the factorization approach. This work was pioneered by Kucera (1980), who gives a general development for time-optimal deadbeat tracking design. He describes the plant as the ratio of polynomials in the delay operator *d*. He uses similar descriptions for the reference input, as a means of defining a given class of inputs, and for a perturbation included to account for arbitrary plant initial conditions. He derives the parameters of a two-degree-of-freedom (feedback plus feedforward) controller from the solutions of two linear polynomial equations.

Following the polynomial approach, Wang and Chen (1986), assume a one-degree-of-freedom (feedback only) controller. Design freedom in their approach arises from allowing the settling time to be longer than the minimum. They consider the idea of optimization of the transient response (zeros) subject to deadbeat tracking constraint. Funahashi and Katoh (1992) use polynomial descriptions to derive a parameterization of all two-degree-of-freedom deadbeat controllers. Also, regarding deadbeat response of the control signal, they show that "input/output" deadbeat control, as they call it, is equivalent to standard "output" deadbeat control provided the plant has no zeros in the region of stability. Peng and Hanus (1993) also follow a polynomial approach in deriving an FIR prefilter design for use when the transient response a of deadbeat control system has too much overshoot or undershoot. In essence, the use of such a prefilter compromises minimum settling time for a smaller transient-response amplitude.

Extensions of the polynomial approach to the multi-input, multi-output (MIMO) case have also appeared in the literature. For example, Eichstaedt (1982) and Wolovich (1983) present variations on a matrix polynomial approach. The former compares the conditions for the existence of the solutions to a closed-loop deadbeat servo problem and an open-loop deadbeat servo problem for arbitrary initial conditions of the plant. The latter derives a general two-degree-of-freedom MIMO controller, and mentions that the solution given is consistent with the internal model principle, already well known for the continuous-time case. Chen *et al.* (1984) use stable rational (not polynomial) matrix factorization approach to derive a general one-degree-of-freedom deadbeat controller for step inputs.

4. Robust deadbeat control

Minamide (1984) presents an algorithm that theoretically provides deadbeat parameter estimation for linear systems in the absence of noise. In the process of on-line identification, the resulting parameter estimates change only $2n$ times. Then, using the converged parameter estimates, a deadbeat control is presented for output tracking. The output error is nonzero at most $2n^2$ times — at n steps for each of the $2n$ times the parameter vector estimate changes. If all the parameter estimate changes occur in the first $2n$ steps, then the deadbeat control provides a settling time of $3n$ steps.

Gopal and Nair (1985) consider the deadbeat regulator problem for multi-input systems, using a state-variable description. They start from a continuous-time plant description, discretized assuming a zero-order hold; then, they derive a means of using the additional freedom available once the poles have been placed at the origin to minimize the closed-loop system sensitivity to parametric uncertainties.

Zafiriou and Morari (1985) consider the SISO set-point tracking problem for discrete-time systems. They review several design methods, and propose a simple generic design method for stable plants that avoids the shortcomings of the other reviewed methods. The generic design method provides deadbeat response, unless the plant has a zero outside the unit circle with positive real part, in which case deadbeat response is compromised to avoid large overshoots or undershoots. In the case of modeling error, robust stability is guaranteed by the use of a parameter (staleness factor) that decreases the bandwidth of the controller. The parameter can also be used to decrease the amplitude of the control input, at the expense of slower response. Apart from its relevance to robust deadbeat control in particular, this paper is quite interesting for its clear and unique presentation of the discrete-time control problem in general.

Zhao and Kimura (1986) present an idea similar to that of Gopal and Nair (1985): Allow non-minimum-time deadbeat response, and use the resulting design freedom to obtain maximal robustness. Their approach is to use the Youla parameterization of all stabilizing controllers, and restrict the parameter to the class that yields the set of all deadbeat controllers. Then, they can choose the parameter so as optimize the L_2 norm of the sensitivity function of the closed-loop system. The longer the settling time allowed, the more robust the system can be made. To achieve maximum robustness, the settling time must be allowed to approach infinity. The authors pursue the same idea further in a follow-up paper (Zhao and Kimura, 1989). They show that a two-degree-of-freedom controller can achieve with finite settling time the robustness that is possible with one-degree-of-freedom controllers only by allowing infinite settling time.

5. Ripple-free deadbeat control

Discrete-time control system models are used for the most part to represent continuous-time systems with sampled measurements and computer-generated (zero-order-hold) inputs. It is important to know whether or not a sampled-data continuous-time system designed to have deadbeat response at the sampling instants will also have deadbeat response in between. Several papers have discussed intersample behavior, or investigated the so-called "ripple-free" deadbeat control.

Crossley and Porter (1974) compare deadbeat control with minimum-time (bang-bang) control in the continuous-time system. They note that, while the time-optimal strategy results in a nonlinear control for which switching surfaces must be determined, the deadbeat control strategy results in a linear feedback that is easily computed. They present an interesting example showing that the deadbeat control compares favorably to the bang-bang control in terms of the settling time and intersample response.

Nagel (1974) derives and analyzes controls that optimize the sampled-data system response, subject to the response being deadbeat at the samples. The optimization involves absolute-value and quadratic integrals of the intersample error, which is analyzed by use of the modified z transform. The optimized deadbeat controls are not ripple-free. By comparison with a ripple-free deadbeat control design, the optimized designs provide a shorter settling time for the sampled error, but allow nonzero intersample error after the discrete-time transient subsides.

Kaczorek (1982) proposes a method for achieving ripple-free deadbeat control: Start with an output-feedback (PD) pole placement for the continuous-time system; choose the closed-loop pole locations to correspond to a model of the continuous-time reference input, in accordance with the internal model principle; then, use a geometric approach to compute a piecewise-constant (discrete-time, ZOH) control to drive the state of the discrete-time system to zero in a finite number of steps. The presence of the continuous-time internal model ensures that the deadbeat response will be ripple-free.

Jetto (1989) uses a polynomial approach to the design of a discrete-time controller that guarantees deadbeat response of the control input as well as the output. (This type of controller is called "input/output deadbeat" by Funahashi and Katoh (1992).) Such a controller avoids the type of intersample ripples that may arise from an oscillating control input.

Zak and Blouin (1993) give a good overview of the ripple-free deadbeat control problem. They contribute to the literature a clear and explicit statement of the internal model principle that determines the existence of ripple-free deadbeat controls.

6. Special topics in deadbeat control

Generalizations of deadbeat control to nonlinear, time-varying, or infinite-dimensional systems have appeared in the literature. Grasselli, Isidori, and Nicolo (1980) study the deadbeat control of bilinear systems, with a particular focus on conditions for controllability to the origin. Grasselli and various coworkers also study the deadbeat control of linear periodic systems (Grasselli and Lampariello, 1981; Grasselli, 1984; Grasselli and Longhi, 1986a,b). Kaczorek (1983a,b) studies the deadbeat control of nonlinear, time-varying systems, as well as 2-dimensional systems.

III. ONE-STEP-AHEAD CONTROL

A. INTRODUCTION TO ONE-STEP-AHEAD CONTROL

1. Standard Derivation

One-step-ahead control is often overlooked in introductory discrete-time control classes, perhaps because of its necessarily large control signals and difficulty of analysis in the z-domain. However, it is easily developed in the time domain, where the plant is described by the difference equation

$$\begin{aligned} &y(n+k) + a_{k-1}y(n+k-1) + \ldots + a_1 y(n+1) + a_0 y(n) \\ &\qquad = b_{n+k-1}u(n+k-1) + b_{n+k-2}u(n+k-2) + \ldots + b_0 u(n) \,. \end{aligned} \tag{12}$$

The idea of one-step-ahead control is to determine the present control signal required to force the output to the desired level in one timestep. Thus in the difference equation above, $y(n+k)$ is the output at the next timestep, whose desired value is $y^*(n+k)$. The present control signal is $u(n+k-1)$. Knowing the present output $y(n+k-1)$ and the desired output at the next step $y^*(n+k)$, and given all of the previous inputs and outputs, it is possible to solve for the present control signal $u(n+k-1)$;

$$\begin{aligned} u(n+k-1) = \frac{-1}{b_{n+k-1}} \{&b_{n+k-2}u(n+k-2) + \ldots + b_0 u(n) \\ &- a_{k-1}y(n+k-1) - \ldots - a_1 y(n+1) - a_0 y(n)\} \,. \end{aligned} \tag{13}$$

This is the one-step-ahead controller.

The representation of one-step-ahead control in the z-domain is rather complicated. Hartley and Sarantopoulos (1991) discuss this representation, and give a block diagram of the one-step-ahead control system. It is shown that the controller only cancels the system zeros, and not the poles. Thus the open-loop plant must be inversely stable to avoid instability. To understand that the poles are not canceled, it should be observed that the denominator dynamics are subtracted out rather than factored out. It is important to note that the closed-loop system always tracks with only a one timestep delay, assuming $b_{n+k-1} \neq 0$. Goodwin and Sin (1984) discuss multivariable one-step-ahead controllers.

2. Historical narrative

The historical narrative for one-step-ahead control is somewhat more convoluted than that for deadbeat control. As such, the discussion is divided into logical subdivisions below.

a) Origins: One-step-ahead Optimal

The main origins of one-step-ahead control lie in optimal control theory. Classically, the optimal control problem requires the determination of the control input that will drive the output of a given system close to some final state while using the minimum amount of energy, fuel, or time. The minimum time to achieve tracking of a given input is of particular interest. The general solution to this problem is a complicated one, that requires the calculus of variations, as discussed in many optimal control texts. In discussing this problem in the final sections of his text, Lee (1964) considers a simplification to the performance measure which uses "a single stage process." Rather than having the final time free, Lee suggests the performance measure

$$J(u(k)) = x^{\mathrm{T}}(k+1)Qx(k+1) + u^{\mathrm{T}}(k)Ru(k), \tag{14}$$

subject to

$$x(k+1) = f(x(k),u(k)). \tag{15}$$

This optimal control problem is easy to solve since it only requires optimization over one timestep, or stage. Taking

$$\frac{\mathrm{d}J}{\mathrm{d}u(k)} = 0, \tag{16}$$

we obtain

$$u(k) = -R^{-1}\frac{\mathrm{d}f^{\mathrm{T}}}{\mathrm{d}u}Qf, \tag{17}$$

which in the linear case gives

$$u(k) = -R^{-1}BQx(k+1). \tag{18}$$

Lee states, "It is emphasized here that this control law is obtained using a single stage process. It is obviously nonoptimal with respect to a multistage process. The only merit lies in its simplicity." In a footnote, he also states, "A single stage process is assumed here to avoid the difficulties involved in solving two point boundary value problems, hence making the problem amenable to real time solutions." This last statement is prophetic in the sense that one-step-ahead control is often associated with real-time implementations of adaptive control. Later in the book, Lee begins the fulfillment of his own prophesy by applying this design strategy to a stochastic problem, and finally combining it with an identification strategy to indeed form an adaptive regulator. He concludes the section by stating, "It should be emphasized here that this configuration is but a conjecture. It is a reasonable thing to do, whether it has any merits remains to be investigated." Then again in concluding the chapter, he states, "It is conjectured that in a closed-loop regulator-type process where the errors are usually small, the cascading of optimal single-stage processes may be made to approximate optimal multistage processes if we choose the criterion with some logic (that is the magnitude of the error, and so forth). Using this simplified philosophy, an optimal closed-loop identification and control process is concocted. Its stability and performance remains to be investigated." Clearly, Lee showed remarkable foresight, as this discussion touches on most of the concepts that are at the heart of one-step-ahead control.

The next major contribution along these lines was due to Astrom (1970) in his seminal text on stochastic control theory. There he describes the idea of minimum variance control theory using an input-output formulation. The concept is essentially the same as that proposed by Lee (1964), that is, minimize the cost in forcing the output to the desired level in one timestep. In this formulation, there is typically no weighting on the control energy used. In a noise-free environment, it is then possible to track the desired reference with a delay of one step. In a stochastic environment, the controller requires (implicitly or explicitly) a prediction model. This then requires the creation of an optimal predictor, which will predict the output with a minimum variance in the prediction error. Using this predictor, it is then possible to design a controller that minimizes the variance of the given system output using a one-step-ahead performance measure. The result is a controller that minimizes the variance of the output from the desired position. Astrom's application to minimizing the variance of paper thickness to allow equal paper production with less pulp usage has become one of the classic success stories of control engineering. This approach is supported further, for unknown systems, in Astrom and Wittenmark (1973).

This approach was further developed by Clarke and Gawthrop (1975) where the input-output approach in a stochastic environment was maintained, but the performance measure was further generalized to include a variety of weightings (P,R,Q) on past inputs (u) and outputs (y), and on future desired positions (w). Their performance measure is

$$J = \mathrm{E}\{[\sum_i (p(i)y(n+k-i)) - \sum_i (r(i)w(n-i))]^2 + [\sum_i (q(i)u(n-i))]^2\}. \tag{19}$$

It should be noted that this can be rewritten more compactly in terms of z-domain (more correctly q^{-1} shift) operator polynomials as

$$\mathrm{J} = \mathrm{E}\{[P(z^{-1})y(n+k) - R(z^{-1})w(n)]^2 + [Q(z^{-1})u(n)]^2\} . \tag{20}$$

Clearly, a wide variety of weightings are possible which lead to several interesting controllers as discussed in Lewis (1986b). For example, (a) $P=R=1$, $Q=0$ gives one-step-ahead control in a deterministic environment, or minimum variance control in a stochastic environment, (b) $P=R=1$, $Q=q$, a scalar constant, gives weighted one-step-ahead control, and (c) $Q=0$, $P(z)$ and $Q(z)$ general polynomials, gives model reference control where the model is R/P.

Clarke and Gawthrop (1975) then go on to derive a self-tuning controller, similar to Astrom (1970) and Lee (1964), using least squares identification. The "self-organizing" control problem is further studied by Saridis and Lobbia (1972) who compare the "per-interval controller" with the "overall-optimal controller." They conclude "the per interval controller appears to be a better choice from a performance standpoint. However, this statement is only a conjecture based on experimental evidence and has not yet been established theoretically." They also conclude that "Using the per-interval controller, the identification converges faster and with less initial overshoot than in the case of the overall-controller."

b) Origins: Inverse Dynamics

Another approach to one-step-ahead control can be found in the early studies of state-space system properties, in particular, the properties of inverse systems. Many authors contributed to this area in the 1960's primarily beginning with Brockett (1965) and continuing through Silverman (1968, 1969), Sain and Massey (1969), and Singh and Liu (1970) among others. The idea of using inverse system properties for control was presented by Godbole and Smith (1972). Basically, the method presented was to cancel the system to be controlled by its inverse in a closed-loop situation. Godbole and Smith say "The concept of precascading a given multivariable plant with its inverse for control purposes is probably not new. However, the evolution of a practical scheme using this concept has become possible only recently as a result of the study of inverses by several authors ... The use of an inverse for control is a rather direct approach in contrast to the usual one employing optimization theory." It should be noted that this approach is given in continuous time.

Martin-Sanchez (1976) presents an adaptive control scheme where "The control block behaves as the 'adaptive' inverse of the process, and has as its input the desired input to the process." Although the first paper that appears in the reference list is (Godbole and Smith, 1972), it is never directly cited in the body of the paper. It would seem that Martin-Sanchez used the Godbole and Smith controller, but converted it to discrete-time first. The result is basically a deterministic one-step-ahead adaptive controller. Martin-Sanchez then goes to great lengths to demonstrate the hyperstability of his algorithm.

c) Origins: Applications

Apparently "out of the blue," another origin in the one-step-ahead control literature appeared. The work was mostly performed by the same collection of individuals, and was based on an attempt to provide real-time control in an application environment. The original reference is Graupe and Cassir (1967) who provide an adaptive control scheme where the controller minimizes at each timestep the cost function

$$C(kT) = \int_{kT}^{kT+T} \{E^{\mathrm{T}}E + \frac{\mathrm{d}E^{\mathrm{T}}\mathrm{d}E}{\mathrm{d}t\;\mathrm{d}t}\}\, \mathrm{d}t\,. \tag{21}$$

This provides a simple and fast controller which "thus avoid[s] slower techniques such as dynamic programming." This approach is further discussed in (Graupe *et al.*, 1968), where the number of necessary variables is reduced.

This approach culminated with the work of Swanick and Sandoz (1970) and Swanick (1972) where the adaptive controller was applied to a "high vacuum system whose dynamics were unknown and which could not be formulated due to their extreme complexity." The one-step-ahead control law was combined with a system identifier to control the plant. Other application oriented problems are discussed, and references to some other applications are given.

d) Unification

Based on the fundamental desire to provide a fast and stable adaptive control technique, the three paths above were unified, primarily in the work of Johnson and Tse (1978). Here a polynomial state-space representation is used along with the deterministic one-step-ahead cost function

$$J(U(k) = \frac{1}{2}\,\{[R(k+1) - Y(k+1)]^{\mathrm{T}}P\,[R(k+1) - Y(k+1)] + U(k)^{\mathrm{T}}QU(k) \tag{22}$$

which has a simple solution. Johnson and Tse state that "A heuristic justification for its use in an adaptive context is that insertion of estimated parameters into a control computation based on a more distant end time would tend to transmit and compound the estimation error." They further state that this one-step-ahead tracking of a given reference "allows tracking of a broad class of trajectories, including the output of nonlinear or distributed parameter models as well as linear, time-invariant lumped-parameter models, the traditional domain of model-following schemes." Unfortunately, "The attractive simplicity of the control input is somewhat offset by limitations on the class of plants that it can stabilize due to its 'shortsightedness.'"

The approach suggested in (Johnson and Tse, 1978) was further extended in (Goodwin *et al.*, 1981), which established global convergence properties of adaptive one-step-ahead control systems. This work was culminated in the outstanding text by Goodwin and Sin (1984). The extensive treatment of both deterministic and stochastic adaptive control in this text was instrumental in the popularization of the entire area of adaptive control and the related concepts. Further unification and popularization of the n-step-ahead and minimum variance control ideas can be found in the last chapters of Lewis (1986a) for the deterministic case and Lewis (1986b) for the stochastic case.

The method has been further extended by Clarke *et al.* (1987a,b) who present generalized predictive control (GPC), which they call an extension to the generalized minimum variance control, developed in (Clarke and Gawthrop, 1975). They state that "The use of long-range prediction and a multi-stage cost in GPC overcomes the problem of stabilizing a non-minimum phase plant with unknown or variable deadtime." The properties of this algorithm are discussed at length in (Clarke and Mohtadi, 1989).

IV. DISCUSSION

A. UNIFICATION

Although both deadbeat and one-step-ahead control have been discussed at length, little discussion of their interelatedness has appeared in the literature. In fact, it appears that most advocates of deadbeat control are unaware that there is a design strategy that gives just as fast a response, yet does not require unstable pole cancellation. On the other hand, the proponents of the one-step-ahead approaches have been aware of deadbeat control, but have not considered the special relationship that it holds with one-step-ahead control. The wall has only recently begun to be breached. Clarke, *et al.* (1987b) show that state deadbeat control is a specific case of their generalized predictive control. They state that when the "costing horizon" is equal to the system order, along with a few other specifics, that GPC is indeed state deadbeat control. This is similar to the earlier result by Yaz and Selbuz (1984) which shows "that the receding-horizon control is a state deadbeat control if the horizon length is taken to be equal to the state dimension." This result is extended by Selbuz and Eldem (1987).

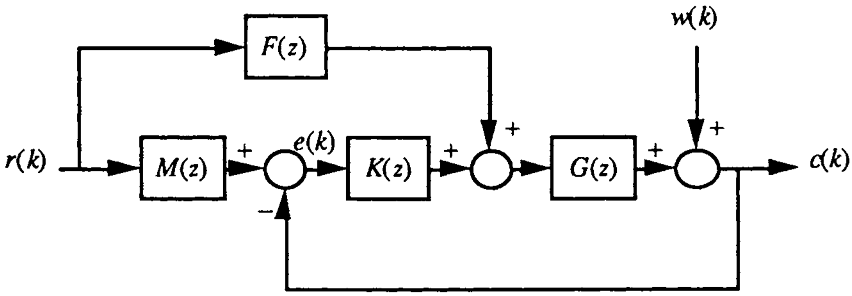


Figure 3. Closed-loop control system with bridged-T controller.

The most complete recognition of one-step-ahead control on the part of the deadbeat community is given by Ichikawa (1989). Interestingly, in an extended introduction dealing with deadbeat control, Ichikawa states "Deadbeat control has been considered by many researchers, but most of the work has been concerned with deadbeat regulators; deadbeat tracking has not been considered as often." Although this is true in the modern era, no mention is given of the fact that originally deadbeat control actually meant deadbeat tracking to a particular input signal. Anyway, in the second paragraph, Ichikawa states, "There is another type of deadbeat control called one-step-ahead control. It has many attractive features, ... Moreover, the reference signal is quite arbitrary. One-step-ahead control is heuristic and the original source is not clear, but a detailed explanation is presented by Goodwin and Sin (1984)." Ichikawa then goes on to study some of the properties of one-step-ahead control.

A further recognition that deadbeat control and one-step-ahead control are related, and share the special property of finite settling time, is given in Hartley and Sarantopoulos (1991). There a special feedback configuration is used to provide a choice of finite settling times to a step input, from one timestep to as many as desired.

B. A NEGLECTED DIRECTION: THE ROBUST DEADBEAT BRIDGED-T CONTROLLER

The concern for robustness permeates the control system literature today, including the work on deadbeat control. In this section, we discuss the bridged-T controller, a configuration that lends itself naturally to robust control design in a deadbeat setting. For some reason, this configuration seems to have been absent from the discussion on deadbeat and one-step-ahead control up until now. It should be noted that the bridged-T controller fits the general form of the two-

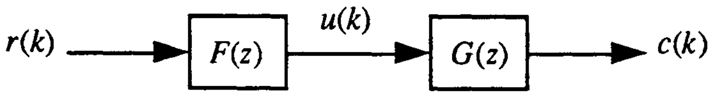


Figure 4. Open-loop control system model.

degree-of-freedom controller, and so has the potential for robust performance demonstrated by Zhao and Kimura (1989).

A closed-loop control system including the bridged-T controller is shown in Figure 3. The plant is represented by $G(z)$, and the controller consists of the combination of $F(z)$, $M(z)$, and $K(z)$. The input $r(k)$ is the reference to be tracked. The input $w(k)$ is a disturbance, which could represent inaccuracy in the model of the plant.

The first controller block, $F(z)$, is an open-loop compensator, designed so that $G(z)F(z)$ provides a suitable response to the reference $R(z)$; see Figure 4. The design of $F(z)$ generally involves the cancellation of the undesirable parts of $G(z)$; so, the plant is assumed to be stable. If the system is to have deadbeat response to a step input at $r(k)$, then $F(z)$ is chosen as

$$F(z) = \frac{n(z)}{z^m} \cdot \frac{1}{G(z)} . \tag{23}$$

Here $n(z)$ may be chosen to keep the transient response of the control sequence $u(k)$ small, or to include the zeros of $G(z)$ outside the unit circle so that $F(z)$ will be stable. Criteria for choosing $n(z)$ may be similar to those found in (Zafiriou and Morari, 1985) for choosing the zeros of the fictitious open-loop compensator $G_c(z)$ discussed there.

The second controller block, $M(z)$, represents the desired control system model. Given the design of $F(z)$, it is chosen as

$$M(z) = G(z)F(z) = \frac{n(z)}{z^m} . \tag{24}$$

The last controller block is the feedback compensator $K(z)$, whose function is to correct for the nonzero errors at $e(k)$. Under ideal conditions, the choices of $F(z)$ and $M(z)$ according to Eqns. (23) and (24) result in $e(k)$=0, and $K(z)$ is not needed. However, in a practical design, $K(z)$ is essential for robustness and disturbance rejection. Further, as the only controller block inside the feedback loop, $K(z)$ is the only one that affects the closed-loop system stability.

The design of $K(z)$ for robustness is a standard problem, and does not depend on the other controller blocks. It is treated, without reference to deadbeat control or the bridged-T controller, in the book by Doyle *et al.* (*Feedback Control Theory*, Macmillan, New York, 1992). For reduction of the error $e(k)$ despite modeling uncertainty or disturbance input, $K(z)$ should be chosen to minimize the weighted sensitivity function

$$S(z) = \frac{1}{1+G(z)K(z)} \cdot H(z)\,, \tag{25}$$

where $H(z)$ is a frequency-weighting transfer function that takes into account the frequency content of the disturbances or modeling errors affecting the system. For improved stability margins (i.e., robustness to multiplicative uncertainty), $K(z)$ should be chosen to minimize the weighted complementary sensitivity function

$$T(z) = \frac{G(z)K(z)}{1+G(z)K(z)} \cdot H(z)\,. \tag{26}$$

The competing objectives of minimizing both $S(z)$ and $T(z)$ can be combined in a single performance index to be optimized.

The advantage of the bridged-T controller is that nominal deadbeat response is achieved first by the design of the blocks $F(z)$ and $M(z)$ outside the loop. Then, independent of the deadbeat design, robustness is achieved separately by the design of the feedback compensator $K(z)$.

C. THE FUTURE

What will become of the areas of deadbeat and one-step ahead control? We see the possibility of interesting future research proceeding along two different lines: the theoretical unification of deadbeat and one-step-ahead control, and the practical resolution of robustness issues. In consideration of the practical issues, it is interesting to repeat the discussion that appears at the end of (Schmidt, 1959). Franklin states that "despite [its drawbacks], the finite settling time design will ... probably remain a practical design for [some] systems...." Schmidt replies that "further research would be desirable to determine ... how to design control systems which have only a small sensitivity with parameter variations." After 35 years and scores of papers, the future of deadbeat control still looks about the same.

BIBLIOGRAPHY

Below are two chronological lists of references. The first list includes those that are discussed in the text above. The second includes papers that are not discussed, but are to some extent related to the topics discussed.

In a project such as this one, it is extremely difficult to include every related piece of work. We extend our apologies to those authors whose work has been inadvertently omitted.

A. CITED REFERENCES

1945

H.W. Bode, *Network Analysis and Feedback Amplifier Design*, Van Nostrand, New York (1945).

L.A. MacColl, *Theory of Servomechanisms*, Van Nostrand, Princeton, 1945.

1947

W. Hurewicz, "Filters and Servo Systems with Pulsed Data," Chapter 5 (pp. 231-261) in *Theory of Servomechanisms*, H.M. James, N.B. Nichols, and R.S. Phillips, eds., MIT Radiation Laborartory Series, vol. 25, McGraw-Hill, 1947.

1950

A. Porter and F. Stoneman, "A New Approach to the Design of Pulse-Monitored Servo Systems," *Proc. IEE*, Pt. II, vol. 97, pp. 597, 1950.

Ya. Z. Tsypkin, "Theory of Intermittent Control Systems" (in Russian), *Automat. Telemech., Moscow*, vol. 11, p. 300, 1950.

1951

D.F. Lawden, "A General Theory of Sampling Servo Systems," *Proc. IEE*, Pt. IV, vol. 98, pp. 31, 1951.

W.K. Linvill, "Sampled-Data Control Systems Studied Through Comparison of Sampling with Amplitude Modulation," *Trans. AIEE*, vol. 70, pp. 1779-1788, 1951.

R. Oldenbourg and H. Sartorius, *Dynamics of Automatic Control Systems*, 2nd ed., Oldenbourg, Munich, 1951.

1952

R.H. Barker, "The Pulse Transfer Function and Its Application to Sampling Servo Systems," *Proc. IEE*, Pt. IV, vol. 99, pp. 302-317, 1952.

C. Holt Smith, D.F. Lawden, and A.E. Bailey, "Characteristics of Sampling Servo Systems," in *Automatic and Manual Control*, Proc. of the Cranfield Conference (1951), A. Tustin, ed., Butterworth Scientific Publications, 1952.

J.R. Ragazzini and L.A. Zadeh, "The Analysis of Sampled-Data Systems," *Trans. AIEE*, Pt. II, vol. 71, pp. 225-234, November 1952.

1954

A.R. Bergen and J.R. Ragazzini, "Sampled-Data Processing Techniques for Feedback Control Systems," *Trans. AIEE*, Pt. II, vol. 73, pp. 236-247, November 1954.

1956

J.E. Bertram, "Factors in the Design of Digital Controllers for Sampled-Data Feedback Systems," *Trans. AIEE*, Pt. II, vol. 75, pp. 151-159, July 1956.

E.I. Jury, "Synthesis and Critical Study of Sampled-Data Control Systems," *Trans. AIEE*, Pt. II, vol. 75, pp. 141-151, July 1956.

1957

E.I. Jury and W. Schroeder, "Discrete Compensation of Sampled-Data and Continuous Control Systems," *Trans. AIEE*, Pt. II, vol. 75, pp. 317-325, January 1957.

1958

E.I. Jury, Sampled Data Control Systems, Wiley, 1958, pp.184, 195-208.

R.E. Kalman and J.E. Bertram, "General Synthesis Procedure for Computer Control of Single and Multiloop Linear Systems," *Trans. AIEE*, vol. 77, p. 602, 1958.

J.R. Ragazzini, and G.F. Franklin, *Sampled-Data Control Systems*, McGraw-Hill, 1958, pp. 148-181.

O.J.M. Smith, *Feedback Control Systems*, McGraw-Hill, 1958.

1959

R.E. Kalman and J.E. Bertram, "A Unified Approach to the Theory of Sampling Systems," *J. Franklin Inst.*, vol. 267, p. 405, 1959.

S.F. Schmidt, "Application of Continuous System Design Concepts to the Design of Sampled Data Systems," *Trans. AIEE*, Pt. II, vol. 78, pp. 74-79, May 1959.

J.T. Tou, *Digital and Sampled-data Control Systems*, McGraw-Hill, 1959.

1960

R.E. Kalman, "On the General Theory of Control Systems," *First IFAC Congr. on Automatic Control*, vol. 4, pp. 481-492, Moscow,1960.

1963

O.J.M. Smith, "Deadbeat Sampled System Direct Synthesis," *IEEE Trans. Automatic Control*, pp. 240-246, July 1963.

1964

R.C.K. Lee, *Optimal Estimation, Identification, and Control*, MIT Press, 1964.

1965

R.W. Brockett, "Poles, Zeros, and Feedback: State Space Interpretation," *IEEE Trans. Automatic Control*, pp. 129-135, April 1965.

J.P. Hyde and C.C. Li, "On Stable Digital Compensators for Minimum Settling-Time of Sampled-Data Control Systems," IEEE Trans. Automatic Control, pp. 115-117, January 1965.

D.P. Lindorff, *Theory of Sampled-Data Control Systems*, Wiley, 1965.

1967

D. Graupe and G.R. Cassir, "Adaptive Control by Predictive Identification and Optimization," *IEEE Trans. Automatic Control*, pp. 191-194, April 1967.

1968

D. Graupe, B.H. Swanick, and G.R. Cassir, "Reduction and Identification of Multivariable Processes Using Regression Analysis," *IEEE Trans. Automatic Control*, pp. 564-567, April 1968.

L.M. Silverman, "Properties and Application of Inverse Systems," *IEEE Trans. Automatic Control*, pp. 436-437, August 1968.

1969

M.K. Sain and J.L. Massey, "Invertibility of Linear Time-Invariant Dynamical Systems," *IEEE Trans. Automatic Control*, vol. AC-14, no. 2, pp. 141-149, April 1969.

L.M. Silverman, "Inversion of Multivariable Linear Systems," *IEEE Trans. Automatic Control*, vol. AC-14, no. 3, pp. 270-276, June 1969.

1970

K.J. Astrom, *Introduction to Stochastic Control Theory*, Academic Press, New York, 1970.

S.P. Singh and R. Liu, "A Survey of Inverse Systems," *Proc. Nat. Elect. Conf.*, pp. 380-384, 1970.

B.H. Swanick and D.J. Sandoz, "Real-time Adaptive Control of Systems Containing Inaccessible State Variables," *Int. J. Control*, vol. 12, no. 5, pp. 839-848, 1970.

1972

S.S. Godbole and C.F. Smith, "A New Control Approach Using the Inverse System," *IEEE Trans. Automatic Control*, pp. 698-702, October 1972.

G.N. Saridis and R.N. Lobbia, "Parameter Identification and Control of Linear Discrete-Time Systems," *IEEE Trans. Automatic Control*, vol. AC-17, no. 1, February 1972.

B.H. Swanick, "A High-speed Deterministic Adaptive Controller," *Int. J. Control*, vol. 15, no. 5, pp. 833-838, 1972.

1973

K.J. Astrom and B. Wittenmark, "On Self Tuning Regulators," *Automatica*, vol. 9, pp. 185-199, March 1973.

1974

T.R. Crossley and B. Porter, "Dead-beat Control of Sampled-data Systems with Bounded Input," *Int. J. Control*, vol. 19, no. 5, pp. 869-876, 1974.

A.L. Nagel, "Three Different Modifications of the Dead-beat Principle in Optimum Control," *Int. J. Control*, vol. 19, no. 6, pp. 1187-1196, 1974.

1975

D.W. Clarke and P.J. Gawthrop, "Self-tuning Controller," *Proc. IEE*, vol. 122, no. 9, September 1975.

B. Porter and A. Bradshaw, "Design of Dead-beat Controllers and Full-order Observers for Linear Multivariable Discrete-time Plants," *Int. J. Control*, vol. 22, no. 2, pp.149-155, 1975.

H. Seraji, "Deadbeat Control of Discrete-time Systems Using Output Feedback," *Int. J. Control*, vol. 21, no. 2, pp. 213-223, 1975.

1976

A. Bradshaw and B. Porter, "Design of Linear Multivariable Discrete-time Tracking Systems for Plants with Inaccessible States," *Int. J. Control*, vol. 24, no. 2, pp. 275-281, 1976.

A. Bradshaw and B. Porter, "Nilpotency Properties of Linear Multivariable Discrete-time Tracking Systems," *Int. J. Control*, vol. 24, no. 4, pp. 573-583, 1976.

J.M. Martin-Sanchez, "A New Solution to Adaptive Control," *Proc. IEEE*, vol. 64, no. 8, pp. 1209-1217, August 1976.

1977

B. Leden, "Output Dead-beat Control - a Geometric Approach," *Int. J. Control*, vol. 26, no. 4, pp. 493-507, 1977.

1978

H. Akashi and H. Imai, "Output Deadbeat Controllers with Stability for Discrete-Time Multivariable Linear Systems," *IEEE Trans. Automatic Control*, vol. 23, no. 6, pp. 1036-1043, December 1978.

C.R. Johnson and E. Tse "Adaptive Implementation of One-Step-Ahead Optimal Control via Input Matching," *IEEE Trans. Automatic Control*, vol. AC-23, no. 5, pp. 865-872, October 1978.

1980

O.M. Grasselli, A. Isidori, and F. Nicolo, "Dead-beat Control of Discrete-time Bilinear Systems," *Int. J. Control*, vol. 32, no. 1, pp. 31-39, 1980.

V. Kucera, "A Dead-beat Servo Problem," *Int. J. Control*, vol. 32, no. 1, pp. 107-113, 1980.

J. O'Reilly, "The Deadbeat Control of Linear Multivariable Systems with Inaccessible State," *Int. J. Control*, vol. 31, no. 4, pp. 645-654, 1980.

1981

G.C. Goodwin, C.R. Johnson, and K.S. Sin, "Global Convergence for One-Step-Ahead Optimal Controllers Based on Input Matching," *IEEE Trans. Automatic Control*, vol. AC-26, no. 6, pp. 1269-1273, December 1981.

O.M. Grasselli and F. Lampariello, "Dead-beat Control of Linear Periodic Discrete-time Systems," *Int. J. Control*, vol. 33, no. 6, pp.1091-1106, 1981.

J. O'Reilly, "The Discrete Linear Time Invariant Time-optimal Control Problem - An Overview," *Automatica*, vol. 17, no. 2, pp. 363-370, 1981.

1982

M. Adachi and H. Akashi, "Minimal Order of Minimal-time Deadbeat Function Observers," *Int. J. Control*, vol. 35, no. 2, pp. 355-365, 1982.

B. Eichstaedt, "Multivariable Closed-loop Deadbeat Control: a Polynomial-matrix Approach," *Automatica*, vol. 18, no. 5, pp. 589-593, 1982.

T. Kaczorek, "Deadbeat Control in Multivariable Linear Systems," *Int. J. Control*, vol. 35, no. 4, pp. 653-663, 1982.

1983

Y. Erol and K. Loparo, "Deadbeat Control of Discrete-time Systems With Singular State Matrix Using Canonical Decompositions," *Automatica*, vol. 19, no. 5, pp. 575-577, 1983.

M.M. Fahmy and J. O'Reilly, "Dead-beat Control of Linear Discrete-time Systems," *Int. J. Control*, vol. 37, no. 4, pp. 685-705, 1983.

T. Kaczorek, "Deadbeat Servo Problem for 2-dimensional Linear Systems," *Int. J. Control*, vol. 37, no. 6, pp. 1349-1353, 1983.

T. Kaczorek, "Deadbeat Control in Multivariable Non-linear Time-varying Systems with Constraints of Control Inputs," *Int. J. Control*, vol. 38, no. 2, pp. 449-458, 1983.

W.A. Wolovich, "Deadbeat Error Control of Discrete Multivariable Systems," *Int. J. Control*, vol. 37, no. 3, pp. 567-582, 1983.

1984

K.J. Astrom and B. Wittenmark, *Computer Controlled Systems*, Prentice-Hall, pp. 203-205, 1984.

B. Chen, C. Chiang, and F. Hsiao, "Deadbeat Controller Synthesis: Multivariable Case," *Int. J. Control*, vol. 40, no. 6, pp. 1207-1213, 1984.

G.C. Goodwin and K.S. Sin, *Adaptive Filtering Prediction and Control*, Prentice-Hall, 1984.

O.M. Grasselli, "Dead-beat Observers of Reduced Order for Linear Periodic Discrete-time Systems with Inaccessible Inputs," *Int. J. Control*, vol. 40, no. 4, pp. 731-745, 1984.

N. Minamide, "Design of a Deadbeat Adaptive Tracking System," *Int. J. Control*, vol. 39, no. 1, pp. 63-81, 1984.

E. Yaz and H. Selbuz, "A Note on the Receding-horizon Control Method," *Int. J. Control*, vol. 39, no. 4, pp. 853-855, 1984.

1985

M. Gopal and P.P. Nair, "On the Design of a Sensitivity-reducing Optimal Dead-beat Controller," *Int. J. Control*, vol. 42, no. 4, pp. 877-886, 1985.

E. Zafiriou and M. Morari, "Digital Controllers for SISO Systems: a Review and a New Algorithm," *Int. J. Control*, vol. 42, no. 4, pp. 855-876, 1985.

1986

G. Cook, "Control of Flexible Structures via Posicast," *Proc. 18th Southeastern Symposium on System Theory*, University of Tennessee, Knoxville, TN, April 7-8, 1986, pp. 31-35.

O.M. Grasselli and S. Longhi, "Output Dead-beat Controllers and Function Dead-beat Observers for Linear Periodic Discrete-time Systems," *Int. J. Control*, vol. 43, no. 2, pp. 517-537, 1986.

O.M. Grasselli and S. Longhi, "Disturbance Localization with Dead-beat Control for Linear Periodic Discrete-time Systems," *Int. J. Control*, vol. 44, no. 5, pp. 1319-1347, 1986.

F.L. Lewis, *Optimal Control*, Wiley, 1986a.

F.L. Lewis, *Optimal Estimation*, Wiley, 1986b.

S. Wang and B. Chen, "Simultaneous Deadbeat Tracking Controller Synthesis," *Int. J. Control*, vol. 44, no. 6, pp. 1579-1586, 1986.

Y. Zhao and H. Kimura, "Dead-beat Control with Robustness," *Int. J. Control*, vol. 43, no. 5, pp. 1427-1440, 1986.

1987

D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalized Predictive Control-Part I. The Basic Algorithm," *Automatica*, vol. 23, no. 2, pp. 137-148, 1987.

D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalized Predictive Control-Part II. Extensions and Interpretations," *Automatica*, vol. 23, no. 2, pp. 149-160, 1987.

H. Selbuz and V. Eldem, "Stabilizing and Deadbeat Properties of Receding-horizon Controllers," *Int. J. Control*, vol. 45, no. 6, pp. 1931-1939, 1987.

1989

D.W. Clarke and C. Mohtadi, "Properties of Generalized Predictive Control," *Automatica*, vol. 25, no. 6, pp. 859-875, 1989.

K. Ichikawa, "Deadbeat Characteristic of One-step-ahead Control," *Int. J. Control*, vol. 49, no. 1, pp. 25-31, 1989.

L. Jetto, "Ripple-free Tracking Problem," *Int. J. Control*, vol. 50, no. 1, pp. 349-359, 1989.

Y. Zhao and H. Kimura, "Two-degree-of-freedom Dead-beat Control System with Robustness: Multivariable Case," *Int. J. Control*, vol. 49, no. 2, pp. 667-679, 1989.

1990

N.C. Singer and W.P. Seering, "Preshaping Command Inputs to Reduce System Vibration," *ASME J. Dynamic Systems, Measurement, and Control*, vol. 112, pp. 76-82, March 1990.

1991

T.T. Hartley and A.D. Sarantopoulos, "Using Feedthrough for Near One-Step-Ahead Control of Inversely Unstable Plants," *IEEE Trans. Automatic Control*, vol. 36, no. 5, May 1991, pp. 625-627.

1992

Y. Funahashi and H. Katoh, "Input/output Deadbeat Control," *Int. J. Control*, vol. 56, no. 3, pp. 515-530, 1992.

1993

Y. Peng and R. Hanus, "An FIR Prefilter Leading to Bounded Overshoot and Undershoot with Application to Deadbeat Control," *Int. J. Control*, vol. 58, no. 2, pp. 459-470, 1993.

S.H. Zak and E.E. Blouin, "Ripple-Free Deadbeat Control," *IEEE Control Systems Magazine*, pp.51-56, August 1993.

B. RELATED REFERENCES

S.S.L. Chang, *Synthesis of Optimum Control Systems*, McGraw-Hill, New York, 1961.

C.A. Desoer and J. Wing, "A Minimum Time Discrete System," *IRE Trans. Automatic Control*, vol. AC-6, pp. 111-125, 1961.

J. Wing and C.A. Desoer, "The Multiple-input Minimal-time Regulator Problem (General Theory)," *IEEE Trans. Automatic Control*, vol. AC-8, pp. 125-136, 1963.

J. Ackermann, "Zeitoptimale Mehrafach-Abtastregelsysteme," *IFAC Symp. Multivariable Control Systems*, Band 1, Dusseldorf, 1968.

J.A. Cadzow, "Nilpotency Properties of the Discrete Linear Regulator," *IEEE Trans. Automatic Control*, vol. AC-13, pp. 734-735, 1968.

J.L. Pokoski and D.A. Pierre, *IEEE Trans. Automatic Control*, vol. 14, pp.199, 1969.

J.A. Cadzow and H.R. Martens, *Discrete-Time and Computer Control Systems*, Prentice-Hall, pp. 247-273, 1970.

J.B. Farison and F.C. Fu, "The Matrix Properties of Minimum Time Discrete Linear Regulator Control," *IEEE Trans. Automatic Control*, vol. 15, pp. 390-391, 1970.

S. Giraud and M. Nougaret, "Hybrid Design Method for Dead-beat Regulators," *Int. J. Control*, vol. 11, no. 6, pp. 987-1003, 1970.

G. Ludyk, Zeioptimale Diskrete Systeme, Habilitationschrift, 1970.

E. Tse and M. Athans, *IEEE Trans. Automatic Control*, vol. 15, pp. 416, 1970.

V. Kucera, "The Structure and Properties of Time-Optimal Discrete Linear Control," *IEEE Trans. Automatic Control*, vol. 16, pp. 375-377, 1971.

V. Prepelita, "La Stabilization des Systemes Lineaires Discrete par Reaction Lineaire," *Rev. Roum. Sci. Techn.-Electrotechn. et Energ.*, vol. 16, pp. 725-737, 1971.

V.W. Eveleigh, *Introduction to Control System Design*, McGraw-Hill, 1972, pp. 457-468.

V. Kucera, "State Space Approach to Discrete Linear Control," *Kybernetica*, vol. 8, pp. 233-251, 1972.

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

C.T. Mullis, "Time Optimal Discrete Regulator Gains," *IEEE Trans. Automatic Control*, vol. 17, pp. 265-266, 1972.

S.P. Battacharyya, *IEEE Trans. Automatic Control*, vol. 18, pp. 381, 1973.

V. Kucera, "Algebraic Theory of Discrete Optimal Control for Single-variable Systems, Part 2. Open-loop Control," *Kybernetica*, vol. 9, pp. 206, 1973.

C.T. Mullis, "On the Controllability of Linear Discrete Systems with Output Feedback," *IEEE Trans. Automatic Control*, vol. AC-18, pp. 608-615, 1973.

A.L. Nagel, *Int. J. Control*, vol. 18, pp. 753, 1973.

B. Porter, Electron. Lett., vol. 9, pp. 83, 1973.

B. Porter, Electron. Lett., vol. 9, pp. 176, 1973.

B. Porter, Electron. Lett., vol. 9, pp. 547, 1973.

D.J. Sandoz, "Optimal Control of Linear Multivariable Systems Based on Discrete Output Feedback," *Proc. IEE*, vol. 120, pp. 1439-1444, Nov. 1973.

J.D. Aplevich and J.H. Anderson, *Electron. Lett.*, vol. 10, 1974.

P. Bruch, "A New Result in Time-optimal Control of Discrete Systems," *IEEE Trans. Automatic Control*, vol. AC-9, pp. 597-598, 1974.

K. Ichikawa, "Synthesis of Optimal Feedback Control Systems with Model Feedback Observers," *Trans. ASME J. Dyn. Sys., Meas., and Cont.*, vol. 96, pp. 470-474, 1974.

V. Kucera, "Algebraic Theory of Discrete Optimal Control for Multivariable Systems," *Kybernetica*, vol. 10, pp. 1, 1974.

V. Kucera, "Algebraic Approach to Discrete Linear Control," *IEEE Trans. Automatic Control*, vol. 20, pp. 116, 1975.

A. Nagata, T. Nishimura, and M. Ikeda, "Linear Function Observer for Linear Discrete-time Systems," *IEEE Trans. Automatic Control*, vol. 20, pp. 401-407, 1975.

B. Leden, "Deadbeat Control and the Riccati Equation," *IEEE Trans. Automatic Control*, vol. 21, pp. 791-792, 1976.

H. Akashi and M. Adachi, "Minimum-time Regulation Problem of Linear Discrete-time Systems," *IEEE Trans. Automatic Control*, vol. 22, pp. 939-942, 1977.

J.F. Donoghue, "A Comparison of the Smith Predictor and Optimal Design Approaches for Systems with Delay in the Control," *IEEE Trans. Industrial Electronics and Control Instrumentation*, vol. 24, no. 1, pp. 109-117, February 1977.

W.R.E. Wouters, "Single Step Versus Multistep Performance Criteria for Steady State SISO Systems," *IEEE Trans. Automatic Control*, pp. 272-274, April 1977.

Jan Yih-Shuh, *Proc. IEEE*, vol. 65, pp. 1728, 1977.

M. Pachter, "An Explicit Pole-Assigning Feedback Formula with Application to Deadbeat Feedback Construction in Discrete Linear Systems," *IEEE Trans. Automatic Control*, vol. AC-22, pp. 263-265, 1977.

B. Leden, "Multivariable Deadbeat Control," *Automatica*, vol. 13, pp. 185-188, 1977.

H. Akashi, H. Imai, and M. Adachi, "On Time-optimal Control of Linear Discrete-time Systems by Geometric Approach," *Int. J. Control*, vol. 28, pp. 853-867, 1978.

K. Ichikawa, "Discrete-time Fast Regulator with Fast Observer," *Int. J. Control*, vol. 28, pp. 733-742, 1978.

H. Kimura, "Deadbeat Function Observers for Discrete-time Linear Systems," *SIAM J. Control Optim.*, vol. 16, pp. 880-894, 1978.

J.E. Koschmann and J.B. Farison, *Int. J. Control*, vol. 28, pp. 649, 1978.

W.H. Kwon and A.E. Pearson, *IEEE Trans. Automatic Control*, vol. AC-23, pp. 479, 1978.

H. Akashi and H. Imai, "Insensitive Observers for Discrete-time Linear Systems," *Automatica*, vol. 15, pp. 641-651, 1979.

D.W. Clarke and P.J. Gawthrop, "Self-tuning Control," *Proc. IEE*, vol. 126, pp. 633-640, June 1979.

K. Ichikawa, "Design of Discrete-time Deadbeat Reduced Order State Observer," *Int. J. Control*, vol. 29, pp. 93-101, 1979.

V. Kucera, *Discrete Linear Control*, Wiley, 1979.

J. O'Reilly, "Observer Design for the Minimum-time State Reconstruction of Linear Discrete-time Systems," *Trans. ASME J. Dyn. Sys., Meas., and Cont.*, vol. 101, pp. 350-354, 1979.

O.A. Sebakhy and T.M. Abdel-Moneim, "State Regulation in Linear Discrete-time Systems in Minimum Time," *IEEE Trans. Automatic Control*, vol. AC-24, pp. 84-88, 1979.

B. Wittenmark and P.K. Rao, "Comments on "Single Step Versus Multistep Performance Criteria for Steady State SISO Systems"," *IEEE Trans. Automatic Control*, vol. AC-24, no. 1, pp. 140-141, February 1979.

M.M. Fahmi, A.A.R. Hanafy, and M.F. Sakr, "On the Discrete Time-optimal Regulator Control Problem," *Information and Control*, vol. 44, pp. 223-235.

D. Jordan and J. Korn, "Deadbeat Algorithms for Multivariable Process Control," *IEEE Trans. Automatic Control*, vol. 25, p. 486 or 586, 1980.

T. Kaczorek, *Int. J. Sys. Sci.*, vol. 11, pp. 411, 1980.

T. Kaczorek, *Int. J. Sys. Sci.*, vol. 11, pp. 423, 1980.

B.C. Kuo, *Digital Control Systems*, Holt, Rinehart, and Winston, 1980, pp. 514-528.

V. Kucera, "Polynomial Design of Deadbeat Control Laws," *Kybernetica*, vol. 16, pp. 198, 1980.

J. O'Reilly, "Comments on "An Explicit Pole-assigning Feedback Formula with Application to Deadbeat Feedback Construction in Discrete Linear Systems"," *IEEE Trans. Automatic Control*, vol. AC-25, pp. 1012-1013, 1980.

Z. Palmor, "Stability Properties of Smith Dead-time Compensator Controllers," *Int. J. Control*, vol. 32, no. 6, pp. 937-949, 1980.

H. Saraji, *Int. J. Control*, vol. 31, pp. 195, 1980.

O.A. Sebakhy and T.M. Abdel-Moneim, *IEEE Trans. Automatic Control*, vol. AC-25, pp. 604, 1980.

M. Sebek, "Multivariable Dead-beat Servo Problem," *Kybernetica*, vol. 16, pp. 442, 1980.

R. Isermann, *Digital Control Systems*, Springer-Verlag, 1981, pp. 122-133, 212-213, 353-354.

T. Kaczorek, *Int. J. Sys. Sci.*, vol. 12, pp. 393, 1981.

H. Kimura and Y. Tanaka, *IEEE Trans. Automatic Control*, vol. AC-26, pp. 1276, 1981.

F.J. Mullin and DeBarbeyrac, "Linear Digital Control," *Trans. ASME: Basic Engineering*, vol. 86, no. 1, pp. 61-66, 1981.

A. Emami-Naeini and G.F. Franklin, *IEEE Trans. Automatic Control*, vol. 27, pp. 176, 1982.

M.M. Fahmy and J. O'Reilly, *IEEE Trans. Automatic Control*, vol. AC-27, pp. 690, 1982.

R. Isermann, "Parameter Adaptive Control Algorithms-A Tutorial," *Automatica*, vol. 18, pp. 513-528, September 1982.

F.L. Lewis, *IEEE Trans. Automatic Control*, vol. 27, pp186, 1982.

R. Rouhani and R.K. Mehra, "Model Algorithmic Control (MAC); Basic Theoretical Properties," *Automatica*, vol. 18, pp. 401-414, July 1982.

M. Schlegel, *IEEE Trans. Automatic Control*, vol. AC-27, pp. 727, 1982.

K.J. Astrom, "Theory and Applications of Adaptive Control-A Survey," *Automatica*, vol. 19, pp. 471-486, September 1983.

M.M. Fahmy and J. O'Reilly, *IEEE Trans. Automatic Control*, vol. AC-28, pp. 125-127, 1983.

K. Janiszowski, *Int. J. Control*, vol. 37, pp. 159, 1983.

Y.F. Shih, *Proc. IEEE*, vol. 130, pp. 119, 1983.

V. Kucera and Sebek, *IEEE Trans. Auto. Control*, vol. AC-29, pp. 719, 1984.

V. Peterka, "Predictor-based Self-tuning Control," *Automatica*, vol. 20, pp. 39-50, January 1984.

J. Ackermann, *Sampled-Data Control Systems*, Springer-Verlag, 1985, pp. 453-458 and other scattered places.

C.H. Houpis and G.B. Lamont, *Digital Control Systems*, McGraw-Hill, 1985, pp. 403-404.

G.F. Franklin and A. Emami-Naeini, *IEEE Trans. Automatic Control*, vol. AC-31, pp. 661, 1986.

G.F. Franklin, J.D. Powell and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Addison-Wesley, 1986, pp. 396-400.

S. Urikura and A. Nagata, "Ripple-free Deadbeat Control for Sampled-data Systems," *IEEE Trans. Automatic Control*, vol. AC-32, pp. 474-482, June 1987.

Y. Zhao and H. Kimura, *Int. J. Control*, vol. 47, pp. 229, 1988.

Y. Zhao and H. Kimura, *Int. J. Control*, vol. 47, pp. 303, 1988.

G.M. Abdel-Nour and R.J. Mulholland, "Algorithm for Deadbeat Tracking," *IEEE Trans. Automatic Control*, vol. 34, no. 7, pp. 773-775, July 1989.

M.W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, Wiley, 1989.

D.J. Trudnowski and Pierre, "A Performance Limit for a Finite-settling-time System Using Look-ahead Control," *IEEE Trans. Automatic Control*, vol. 34, pp. 760-763, 1989.

J. Van de Vegte, *Feedback Control Systems*, 2nd edition, Prentice-Hall, 1990, pp. 292-295.

T.T. Hartley and G. Cook, "One-Step-Ahead Control with Improved Sensitivity Robustness," *IEEE Trans. Industrial Electronics*, vol. 39, no. 3, June 1992, pp. 268-272.

M. Halpern, "Preview Tracking for Discrete-Time SISO Systems," *IEEE Trans. Automatic Control*, vol. 39, no. 3, pp. 589-592, March 1994.

This Page Intentionally Left Blank

Discrete-Time LQR Techniques in the Control of Modern Canals

Andre Garcia
Hewlett-Packard Company
San Diego, CA 92127-1899

Mont Hubbard
Department of Mechanical and Aeronautical Engineering
University of California, Davis
Davis, CA 95616-5294

I. INTRODUCTION

The operation of modern canals is quite complex--a hierarchical control system is typically employed to minimize operational expenses and to schedule and regulate the flow of water. In this work, we will address the problems of changing the flowrate of water throughout the canal rapidly while controlling the associated surge magnitude and regulating the flow condition. The motivation for these efforts is to reduce water waste by accurately matching supply with demand.

A variety of methods have been proposed for approaching the problem of controlling open channel flow. Early work on the design of feedback controllers for check gates was done by Shand [1] using classical control theory. Buyalski *et al* [2], also using classical frequency response methods, investigated the stability of closed-loop level controllers. Balogun *et al* [3, 4] and Hubbard *et al* [5] applied the linear quadratic regulator (LQR) technique to open channel flow control using a linearized, spatially discretized version of the St. Venant Equations. Predictive control strategies based on simplified flow models have also been investigated [6, 7, 8]. The latter works do not address the problem of transient wave magnitude control.

In this paper, we will address three major issues faced in the application of the LQR theory to the regulation of large flow transitions in multi-pool canal systems. These are: the development of an accurate yet simple linear model of the flow dynamics, a physically meaningful method of performance index selection, and the selection of reference inputs which allow control over transient wave magnitude. The model used to develop the regulation algorithm is a time and space discretized approximation of the St. Venant Eqs. The dynamic response of the wave equation, which is easily developed analytically, is used as a guideline for developing the penalty function coefficients. The wave equation is also used to develop reference inputs for large flow transitions so that transient wave magnitudes may be controlled. A simulation of a low order linear regulator controlling a much higher order nonlinear model of the flow dynamics is used to demonstrate the performance and robustness of the design.

II. OPEN CHANNEL TRANSIENT MODELS

A. THE ST. VENANT EQUATIONS

Shown in Fig. 1 are longitudinal and cross-sectional schematics of a typical canal. Pools typically have trapezoidal cross sections and adjacent pools are separated by check gates, by which the overall flow is regulated. The coordinated control of the gate openings while limiting surge magnitudes is the subject of this work.

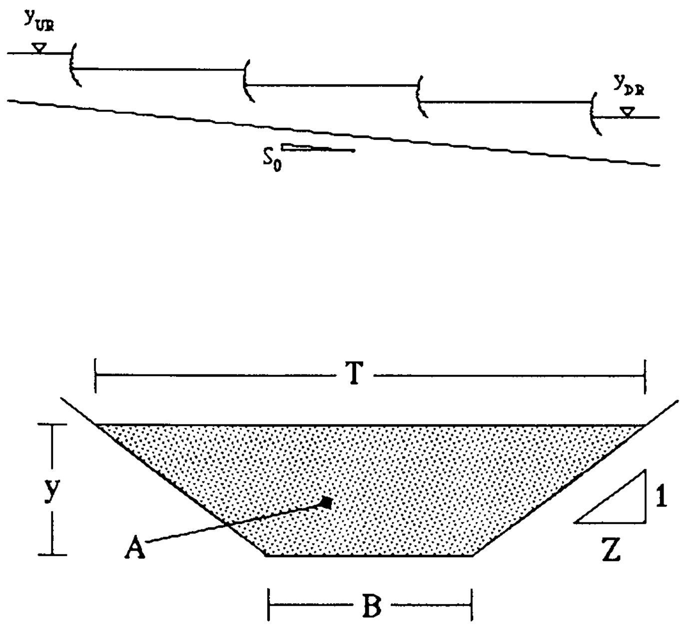


Figure 1 (a) Schematic representation of multi-pool canal. S_O, bottom slope; y_{UR}, upstream reservoir depth; y_{DR}, downstream reservoir depth. (b) Cross-sectional parameters. T, top width; y, depth; A, cross-sectional area; B, bottom width; Z, side slope.

The one-dimensional equations for gradually varied, unsteady flow in a prismatic channel are:

$$A\frac{\partial V}{\partial x} + VT\frac{\partial y}{\partial x} + T\frac{\partial y}{\partial t} + q = 0 \tag{1}$$

$$\frac{1}{g}\frac{\partial V}{\partial t} + \frac{V}{g}\frac{\partial V}{\partial x} + \frac{\partial y}{\partial x} = S_0 - S_f + D_l \tag{2}$$

where y = depth of flow normal to the bottom; V = average (with respect to the cross-sectional area) flow velocity; t = time; x = distance along the channel; A = cross-sectional area; T = water surface width; S_0 = bottom slope; S_f = friction slope (determined from the Manning equation); q = lateral outflow per unit length along the channel; D_l = outflow momentum transport factor; and g = gravitational acceleration (see Fig. 1).

They are derived by integrating the point forms of the equations of motion and continuity over a cross section perpendicular to the flow. The assumptions made in deriving Eqs. (1) and (2) and their range of applicability are discussed thoroughly by Strelkoff [9].

Flow in the canal is controlled by underflow checkgates (see Fig. 2).

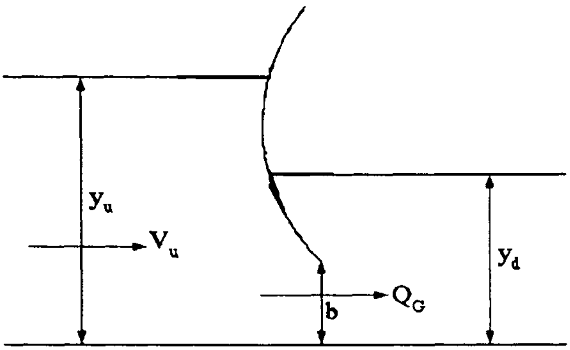


Figure 2 Schematic representation of a check gate. Q_G, gate discharge; b, gate opening; y_u, upstream depth; V_u, upstream velocity; y_d, downstream depth.

The system model will require relationships which describe the influence of the gates on the flow dynamics. The discharge relationship used here for an underflow gate is [10]

$$Q_G = C_D b B_G \sqrt{2g(y_u - y_d) + V_u^2} \tag{3}$$

where Q_G = the under gate discharge; C_D = the discharge coefficient ($C_D < 1$ since there is a head loss as the flow passes through the gate): b = the gate opening; B_G = the gate bottom width: y_u = the depth upstream of the gate; y_d = the depth downstream of the gate; and V_u = the velocity upstream of the gate. Canals considered here are bounded at the upstream and downstream ends by reservoir conditions--i.e. constant depth levels. Special cases of Eq. (3) are used for the upstream and downstream gates.

Sometimes (usually in open loop computations of reference trajectories) a simpler discharge relationship than Eq. (3) is used to model the influence of the gates. In this direct discharge model (called the "alternate model") the gate discharge Q_G is specified directly, rather than the gate opening b. The advantage of the simpler relationship is that it is more linear and therefore more accurate than a linearized version of Eq. (3) for large flow transitions. The use of these two discharge relationships should be clear from the context in what follows.

B. LUMPED MODELS FOR REGULATOR DESIGN

The equations of motion for open channel flow, often referred to as the St. Venant equations, are a coupled set of first order non-linear hyperbolic partial differential equations (PDE). The controlled object--the canal--is thus a distributed parameter system (DPS). In developing control schemes for a DPS, we must decide whether to use the PDE's as our model of the plant or something more tractable. The authors admit to being unfamiliar with any technique for deriving control algorithms directly from a nonlinear distributed parameter model of a system. In this section, we will concentrate on the development of a lumped model which is later used for regulator design.

The considerable DPS control literature [11] addresses the problem of converting a DPE model into a lumped model [12]. Several common reduction techniques--space quantization and eigenfunction expansion, both continuous time methods, and time and space quantization, a discrete time method--were considered.

Balogun *et al.* [4] employed the space quantization technique. The method consists of approximating the spatial derivatives by finite differences and thus transforming the PDE into a set of ODE's. Results of Balogun's [3] accuracy studies of a linearized, space discretized model were largely inconclusive as there was no standard of comparison, although it was noted that realistic wave-like disturbance propagation was reproduced.

For a certain special case, the convergence of the eigenvalues of the linearized, space-quantized model may readily be studied. Linearizing Eqs. (1) and (2) about a nominal equilibrium state and assuming a trapezoidal cross-section ($T = B + 2zy$) gives (the subscripts "x" and "t" denote partial differentiation)

$$A_0 dV_x + V_0 T_0 dy_x + T_0 dy_t + (T_0 V_{x0} + 2zV_0 y_{x0}) dy + T_0 y_{x0} dV + dq = 0 \tag{4}$$

$$\frac{1}{g} dV_t + \frac{V_0}{g} dV_x + dy_x + \frac{1}{g} V_{x0} dV = dD_l - dS_f \tag{5}$$

where each dynamic variable is expressed as the sum of a nominal and a perturbed component, i.e. $V = V_0 + dV$ respectively. If the nominal velocity, bottom slope, and outflow are all zero then Eqs. (4) and (5) reduce to the simple wave equation:

$$\frac{gA_0}{T_0} dV_{xx} - dV_{tt} = 0 \tag{6}$$

If a single pool is considered with boundary conditions $V(0) = V(L) = 0$, the oscillatory frequency of the i^{th} mode is given by:

$$f_i = i\pi \frac{c}{L};\ i = 0,\ 1,\ 2, \ldots \tag{7}$$

where the wave speed c, identified from Eq. 6 is

$$c = \sqrt{\frac{gA_0}{T_0}} \tag{8}$$

Models of order 11, 21, 41, and 81 were created to investigate the spectral convergence of a central difference space-quantized model. The ratios of the first five oscillatory frequencies computed from the lumped models to the fundamental frequency ($\pi c/L$) are plotted in Fig. 3 vs. the lumped model order. For a particular model the error becomes larger for higher frequency modes and the error in all frequencies is reduced as the model order is increased. Also, the computed frequency is always less than the actual frequency.

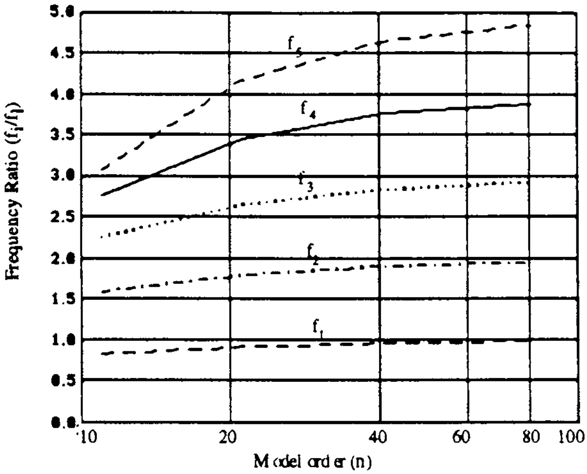


Figure 3 Ratio of first five modal frequencies computed from space-quantized model to the fundamental frequency ($\pi c/L$). As the model order increases, the frequency ratios approach the series 1, 2, 3, ...

Figure 4 shows the convergence of all the oscillatory frequencies as the model order is increased.

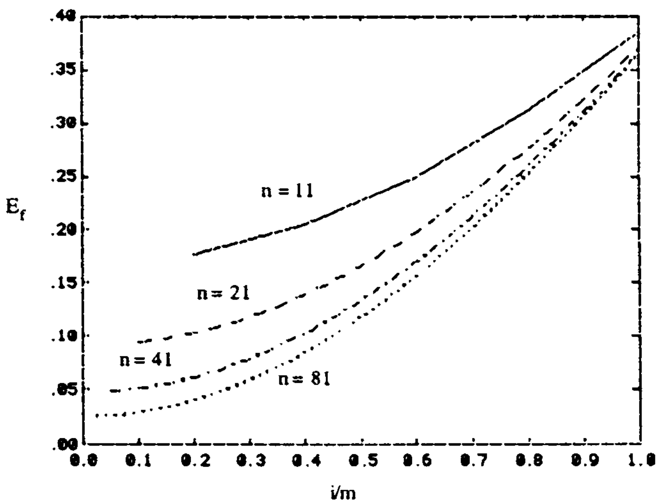


Figure 4 Normalized error ($E_f = (f - f_c)/f$; f_c is the computed frequency) in entire space-quantized model spectra at various discretization levels. The independent variable is the modal index divided by the total number of oscillatory modes.

The abscissa in this case is the modal index, i, divided by the total number of oscillatory modes in the model, m. For an m-mode model, the largest modal index is m, and the smallest modal index for any order model is one, so: $1/m \leq i/m \leq 1$. Each curve on the graph represents the errors in all modal frequencies of a particular model. Note that the slopes of the curves are small for small i/m. This indicates that the errors in the low frequency range of the model are nearly constant. Note that the frequency errors in the lowest frequency modes seem to be about 1/m.

Apparently few of the modal frequencies are accurately represented. Many of the modes of a space-quantized model are poor approximations of the original PDE's modes. Including the inaccurate modes increases the model's complexity without contributing to its accuracy. (Strictly speaking these results are only accurate for the wave equation; but we expect them to be representative of error magnitudes at non-zero flow and with non-zero bottom slopes.) Since the amount of time required for optimal control computations is exponentially related to the model order, any modeling inefficiency is undesirable. The inefficiency of a space-quantized model seems intolerable.

Eigenfunction expansion, another method of creating a continuous time lumped model, requires computing the eigenvalues and eigenfunctions for the linearized PDE. The difficulty in deriving the eigenvalues and eigenfunctions from the linearized St. Venant Eqs. lies in the fact that the coefficient functions of the linearized PDE are known only numerically. However, numerical determination of the eigenvalues and eigenfunctions is straightforward; they may be approximated by the eigenvalues and eigenvectors of the space-quantized model. A suitably large space-quantized model is constructed (to get desired accuracy), and the inaccurate frequencies and modes are truncated from the model. A drawback of this method is that the time delay, a characteristic of hyperbolic systems, is not modeled well.

Lumping techniques which involve both time and space quantization have already been developed for simulating transient conditions in open channels. Their employment with the aid of a digital computer is a routine practice. The stability, accuracy, and applicability of the various techniques have been the subject of a great deal of research [13, 14, 15]. The conclusions of this research indicate that the method of characteristics is well suited to transient conditions encountered in man-made canals because of the regular geometry, relatively small surge magnitudes, and rapid nature of the flow changes.

A linearized method of characteristics (MOC) model was considered best for application of lumped parameter control theory for a number of reasons. It is well known that the method of characteristics is the only proper (and accurate) way to deal with boundary conditions numerically [13]. This is of paramount importance in a control study as the form of the model will influence the optimal gate motions--the more accurately the boundary conditions are handled in the model, the more accurate will be the specification of optimal gate motions. The spectrum of the MOC model exactly matches that of the wave equation when the linearization is performed about zero flow and the bottom slope is zero for any model order. It is recognized that this match deteriorates (because of interpolations) for conditions other than ideal [16]. But, for large wavelength $\lambda / \Delta X >> 1$), small amplitude ($\Delta y/y << 1$), low velocity ($V/c << 1$) transients encountered in typical systems with small bottom slopes ($LS_0/y << 1$) and small friction slopes ($\Delta t g S_f/(2V) << 1$) the model will be sufficiently accurate. (Δx and Δt are the space and time grid spacing of the MOC model, and c and L are the wave speed and overall pool or channel length.

C. METHOD OF CHARACTERISTICS MODEL FOR SIMULATION AND REGULATOR DESIGN

The method of characteristics is a well known analytical procedure for transforming a set of hyperbolic PDE's into a set of ODE's. The ODE's may subsequently be transformed into a set of difference equations through numerical integration and interpolation. The particular interpolation and integration scheme is a matter of preference and the form of the resulting model is, in general, nonlinear and implicit.

$$\mathbf{F}(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_{k+1}) = \mathbf{0} \tag{9}$$

The development of the model used here, by first order integration and first order interpolation, closely follows that in Wiley *et al* [15]. A nonlinear MOC model is used to model the flow dynamics in all simulations presented here. However a linear model is needed for LQR design. The first order method chosen simplifies the linearization; use of "the method of specified time intervals" allows us to create a semi-standard discrete time state equation,

$$\mathbf{x}_{k+1} = \Phi \mathbf{x}_k + \Gamma \mathbf{u}_{k+1} \tag{10}$$

whereas the form commonly treated in the lumped control theory literature is

$$\mathbf{x}_{k+1} = \Phi \mathbf{x}_k + \Gamma \mathbf{u}_k \tag{11}$$

Also of note is the fact that Φ is not of full rank because both depth and velocity states have been included at the boundaries and they are not independent (because of the gate equation - Eq. 3). The non-standard form of (10) is also a consequence of the boundary conditions which can immediately affect states adjacent to the boundary. The linearized MOC model is somewhat more difficult to develop than the continuous time models. Some of the elements of Φ and Γ are cumbersome to express explicitly because of the implicit nature of Eq. 9.

The discrete time state equations (10) derived from the first order MOC can be made to conform with the standard by simply changing the input. Instead of using the gate positions (discharges) as input, the gate velocities are used. If the gate velocities are constant throughout the time step Δt, the gate position vector at t_{k+1}, $\mathbf{b}_{k+1}$ is given by

$$\mathbf{b}_{k+1} = \mathbf{b}_k + \Delta t \mathbf{r}_k \tag{12}$$

where $\mathbf{r}_k$ is the vector of gate velocities from t_k to t_{k+1}. When this equation is substituted into (10), we have

$$\mathbf{x}_{k+1} = \Phi \mathbf{x}_k + \Gamma \mathbf{b}_{k+1} = \Phi \mathbf{x}_k + \Gamma \mathbf{b}_k + \Delta t \Gamma \mathbf{r}_k \tag{13}$$

Equations (12) and (13) together are the state equations. The augmented state now includes the gate positions. We could probably work with Eq. (10) and its non-standard control index, but this model has some advantages over the gate position input model. The gate positions are now known at all times and also it will be possible to control (penalize) the magnitude of the gate velocities and thus prevent gate motions which are not practically feasible due to finite gate motor speeds.

III. WAVE EQUATION DYNAMICS

A number of modern control techniques may now be applied to flow control in canals using the lumped linearized MOC model of open channel flow dynamics. In this section, some insights into the peculiar dynamics of this system are explored. Using the simple wave equation, two fundamental concerns are addressed--the amount of time required to change the flow condition in a channel and the relationship between the amount of time to effect a flow change and the magnitude of generated surges. The important case of making a transition between one steady flow condition and another is considered. The lumped MOC model retains some important features of the hyperbolic governing equations; the state of the lumped model can not be completely changed in less than the time required for a disturbance to travel the length of a single pool. The wave equation can be used to predict that the maximum surge magnitude occurs when the flow condition is changed in the minimum time. As the transition time is lengthened by multiples of the minimum time, the surge magnitude is proportionately reduced.

A. MINIMUM TIME CONTROL

The minimum transition time between arbitrary initial and final conditions may be obtained graphically using the method of characteristics. Fig. 5 shows the characteristic lines in the x-t plane for a single pool. In Fig. 5a the behavior in the lower shaded region, the zone of quiet, is controlled by the initial condition.

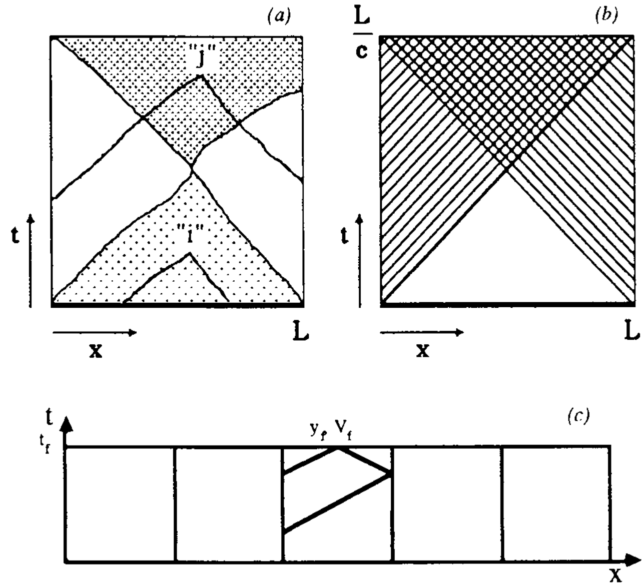


Figure 5 Domains of dependence. (a) General one-dimensional hyperbolic equations. Both characteristics which intersect at point "i" emanate from the initial condition. Hence the state at "i" and the entire lower shaded region cannot be affected by moving the gates located at x = 0 and x = L. Both characteristics which intersect at point "j" emanate from the boundaries. A desired state at "j" can be achieved by proper control of the conditions at the boundary points from which the intersecting characteristics emerge. Similarly, the state in the entire upper shaded region can be controlled. (b) For the wave equation, all characteristics are straight and the earliest time at which the state along the entire pool is controllable is t = L/c. (c) In a multiple pool system, since two gates can not be dedicated to the control of a single pool, the minimum time control takes longer. If one gate is used to control each pool, the state along the entire canal can be controlled in at least the time it takes for a disturbance to travel up and down the longest reach by reflecting signals as shown.

The behavior in the upper shaded region is determined entirely by the boundary conditions for $t > 0$. For the simple wave equation (Fig. 5b), the characteristic lines are straight, and it is seen that for $t > L/c$ the state is determined entirely by the boundary conditions for $t > 0$.

Derivation of the minimum time to accomplish an arbitrary state transition from the state space model [17] is more tedious than the direct graphical technique but yields the same result independent of the model order.

The general result for a series of p equal length pools bounded by r (= p + 1) gates is

$$t_{min} \geq \frac{2pL/c}{r} \tag{14}$$

t_{min} is equal to the expression on the right side of (8) when p is one or three, but it can not be equal when p is two, four, or five. The general relationship has not been proven by the authors, but it appears that the equality only holds when 4p/r is an integer. This is only the case for p equal to one or three.

Consider a series of channels as shown in Fig. 5c. The final condition at t_f may be controlled if t_f is greater than or equal to the time it takes for a disturbance to travel up and down the longest pool. In such a case, each of the two characteristics intersecting at a point on the time line, t_f, will reach one point on a single boundary where the boundary condition may be controlled independently to affect that characteristic. The two boundary conditions, one for each characteristic, may be chosen so that an arbitrary desired state, y_f and V_f, is achieved at t_f. If, for example, the upstream gate in each pool is used to control the final state, then the downstream most gate can be moved arbitrarily (as long as its final position is compatible with the steady state flow). Referring to the wave equation model of a series of equal length pools, we have

$$2L/c \geq t_{min} \geq \frac{2pL/c}{p+1} \tag{15}$$

The lower bound seems rarely to be achieved, but the general case which would apply to canals with various pool lengths (if L is the longest pool), is that the minimum time to accomplish an arbitrary state transition is 2L/c.

The fundamental result that complete state control is not possible in a single pool until both the positive and negative waves traverse the pool, which seems obvious in hindsight, is important enough to repeat. The implication is that there are limits to what can be achieved with any kind of control--open or closed-loop. Insuring that the state variable model is subject to the same limitations as the real system (even in the highly idealized wave equation example) will prevent us from drawing false conclusions. For example, Balogun [3] concluded that a lumped continuous time canal model was "controllable." A continuous time system is controllable if an arbitrary change in state can be brought about in an arbitrary (but finite) time interval. A canal controlled by check gates is not controllable in the continuous time sense. Notice that the time delays inherent in this hyperbolic system are naturally modeled by the simple discrete time state equations and therefore that the state space model predicts performance limitations inherent in a real canal.

B. SURGE/TRANSITION TIME TRADE-OFF

An important constraint in canal operations is the allowable deviation in water levels (another constraint is the allowable rate of water level deviation). Using the wave equation as a model of fluid behavior, the magnitude of surges which accompany flow transitions is examined here. A great deal of insight into the trade-off between the duration of a flow transition and the magnitude of the accompanying surge may be obtained by studying the simple wave equation using the method of characteristics. This insight will help in the selection of penalty matrices used for compensator gain calculations and in constructing reference inputs for the control system when we would like to make a large flow transition and meet operating constraints. The analysis will point out one reason why it is not always possible to use the minimum time control discussed previously.

The method of characteristics for the simple wave equation reduces to

$$\begin{aligned} V_P - V_R + \frac{g}{c}(y_P - y_R) &= 0 \\ \Delta x &= c\Delta t \end{aligned} \tag{16}$$

$$\begin{aligned} V_P - V_S - \frac{g}{c}(y_P - y_S) &= 0 \\ \Delta x &= -c\Delta t \end{aligned} \tag{17}$$

where V_P and y_P are the velocity and depth deviations at point 'P', and the 'S' subscripted variables apply to the minus characteristic and the 'R' subscripted variables come from the positive characteristic (see Fig. 6).

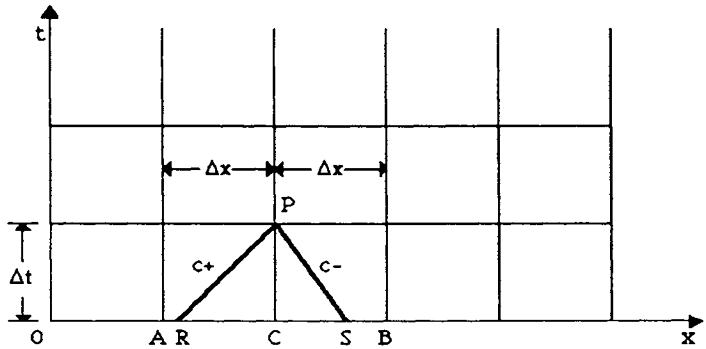


Figure 6 The computation grid for the method of specified time intervals.

First let us determine the surge magnitude for the minimum time flow transition. At t = 0, the flow velocity is zero, and at t = L/c, the flow velocity is V_f and the depth perturbation is zero. By considering characteristic pairs which travel from the quiescent initial condition, reflect off of the boundaries and then reach the steady final condition. For 0 < t < L/c we have

$$V(0,t) = \frac{g}{c}y(0,t)\ ;\ V(L,t) = -\frac{g}{c}y(L,t) \tag{18}$$

$$V_f - V(0,t) = \frac{g}{c}y(0,t)\ ;\ V_f - V(L,t) = -\frac{g}{c}y(L,t) \tag{19}$$

and it follows that

$$y(0,t) = \frac{cV_f}{2g}\ ;\ y(L,t) = -\frac{cV_f}{2g} \tag{20}$$

A positive surge (assuming $V_f > 0$) of magnitude $V_f\,c/2g$ enters from the left and a negative surge of the same magnitude enters from the right. Both of these depth surges are accompanied by positive (assuming $V_f > 0$) velocity surges of magnitude $V_f/2$. At t = L/2c, the surges collide; the depth surges cancel and the velocity surges add to V_f. The zero depth wave and the velocity wave of magnitude V_f propagate from the center of the pool to the boundaries until at t = L/c uniform flow is achieved (see Fig. 7).

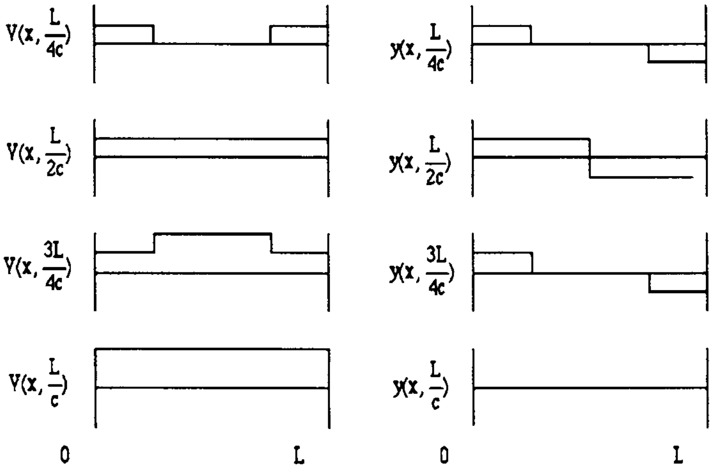


Figure 7 Single pool transient response for minimum time control using a wave equation model. The velocity (V) and depth (y) distributions are shown.

Now, because the wave equation is linear (or because it is a perturbation equation), this solution could be added to a constant uniform flow at V_i. In this case, the magnitude of the surge is

$$h_s = \frac{c(V_f - V_i)}{2g} \tag{21}$$

Since the final flow condition is uniform, we may proceed from uniform flow at V_f to uniform flow at $2V_f$, with the same surge magnitude, $V_f c/2g$. Thus the time for the flow transition in the two step process from 0 to $2V_f$ is $2L/c$ and it is accompanied by a constant surge magnitude of $V_f c/2g$ at the boundaries. If the flow transition from 0 to $2V_f$ were accomplished in the time interval L/c, the surge magnitude would be $V_f c/g$. In general, when the surge magnitude is constant at the boundaries

$$h_s = \frac{\Delta V L}{2g t_T}; \quad t_T = n\frac{L}{c}; \quad n = 1, 2, 3, \ldots \tag{22}$$

A simple technique has been used to generate a "family" of flow transitions. This simple analysis points out the principal, and rather intuitive, trade-off involved in accomplishing a flow transition from one steady state to another. Clearly, from (22), the maximum surge is inversely proportional to the flow transition time (and proportional to the change in the mean flow velocity).

Thus, as one would expect, the price of a rapid transition is a large surge. Equation (22) provides a rough estimate of the time, t_T, needed for a flow transition when surges must be limited to a given magnitude, h_S. In some cases the surge for the minimum time transition may be intolerable.

A problem with the constant surge magnitude family of flow transitions is that they result in instantaneous depth changes (infinite drawdown rates) throughout the pool. If the flow at the boundary is varied in a ramp-like fashion (V(0, t) = V(L, t) = at, t > 0), the drawdown rate may be controlled. Applying the method of characteristics as previously we have for $0 \leq t \leq L/c$

$$y(0, t) = \frac{c}{g}\alpha t\ ;\ y(L, t) = -\frac{c}{g}\alpha t \tag{23}$$

and for $L/c \leq t \leq 2L/c$

$$y(0, t) = \alpha\frac{c}{g}(2\frac{L}{c} - t)\ ;\ y(L, t) = \alpha\frac{c}{g}(t - 2\frac{L}{c}) \tag{24}$$

Thus y(0, 2L/c) = y(L, 2L/c) = 0. In fact it is easily verified that y(x, 2L/c) = 0 and V(x, 2L/c) = 2αL/c for $0 \leq x \leq L$ (see Fig. 8).

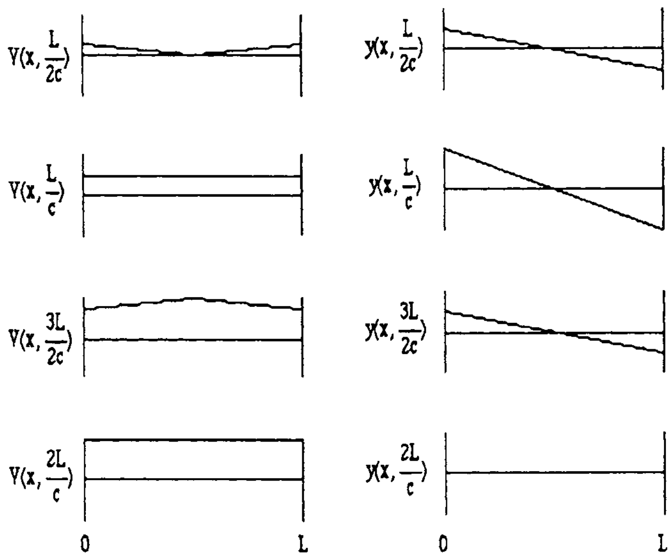


Figure 8 Single pool transient response for ramp control input using a wave equation model. The velocity (V) and depth (y) distributions are shown.

If the boundary flow velocities are maintained at $V_f = 2\alpha L/c$ a new equilibrium is attained. Notice that the drawdown rate magnitude, which is maximum at the boundaries, is constant throughout the flow transition.

$$\frac{dh_s}{dt} = \alpha\frac{c}{g} = \frac{c^2 V_f}{2gL} \tag{25}$$

By superposing solutions, as in the constant surge magnitude case, the general relationship between the drawdown rate magnitude and the transition time, t_T, for the family of flow transitions in which the drawdown rate magnitude is constant is

$$\frac{dh_S}{dt} = \frac{2\Delta VL}{gt_T^2} \tag{26}$$

Also note that in this case the maximum drawdown is

$$h_S = \frac{\Delta VL}{gt_T}; \quad t_T = 2n\frac{L}{c}: \quad n = 1, 2, 3, \ldots \tag{27}$$

Equation (26) reveals how transition times are limited by particular facilities. Substituting $\Delta V = \Delta Q/A$ and rearranging

$$t_T \geq \sqrt{\frac{2\Delta QL}{gA(dh_s/dt)_{max}}} \tag{28}$$

ΔQ, the discharge change, is determined by water needs. Thus the important facilities parameters are: L, the check gate spacing, A, the wetted canal cross-section, and $(dh/dt)_{max}$, the maximum drawdown rate. In some situations, the drawdown rate alone is the limiting constraint. For example, since the drawdown rate limit is constant during the first hour of a transient period for the CSWP, any constant drawdown rate magnitude transition taking less than two hours is acceptable if (28) is satisfied (because the drawdown rate will change sign half way through the transition). The situation becomes more complicated when the flow transition takes longer. The largest flow transition limited by the simple rate constraint for the CSWP is thus 220 m^3/s ($t_T = 6L/c < 7200$ s, L = 7300 m, c = 7.6 m/s, A = 230 m^2, $(dh_s/dt)_{max} = 4.3(10^{-5})$ m/s (6 in. per hr.)).

C. GATE STROKING LIMITATIONS

The results of gate stroking applications may be evaluated by comparing the computed drawdown and drawdown rate with the ideal minimum values (22 and 26). Total normalized drawdown ($h_s g t_T/(\Delta VL)$) of 1.04 and 1.05

for flow initiation to 100 m^3/s and 170 m^3/s respectively are shown in Bodley *et al* [18]. Normalized drawdown rates ($dh_s/dt\ gt_T^2/(\Delta VL)$) of 4.6 and 3.5 are shown for the same flow initiations. Both the drawdown and the drawdown rates are thus larger than the minimum normalized values of 0.5 and 2 respectively. Interestingly, the computed drawdown in these gate stroking applications are about equal to the drawdown predicted when the drawdown rate magnitude is constant (($h_s g t_T/(\Delta VL) = 1$). But the drawdown rate magnitude is about twice what it theoretically could be ($dh_s/dt\ gt_T^2/(\Delta VL) = 2$).

The gate stroking procedure detailed in Bodley *et al* [18] is not an optimal way of controlling the depth variations during a transient. Depth variations cannot be controlled during the periods when the characteristics which emanate from one controlled boundary travel to the other boundary since their arrival is not anticipated in the method. Examining Bodley's results one finds that within these periods the drawdown rate is the greatest and the surge peaks.

When we try to apply the previous analysis to the problems of determining the penalty matrices used for gain calculations and creating reference input to the regulator, our success depends of course on the degree of similarity between the full St. Venant equations for flow in a canal and the wave equation. To demonstrate the accuracy of surge heights predicted by the wave equation for realistic canal operations, several simulations were performed using the minimum drawdown rate strategy proposed above. Canal dimensions (from Bodley *et al* [18]) are typical of the California Aqueduct (see table 1) and the flow transitions simulated are maximum flow initiation (0-180 m^3/s) and flow arrest (180-0 m^3/s).

Table 1 California Aqueduct representative parameters taken from Bodley *et al.* (1978)

Parameter	Symbol	Value
Bottom width	B	12.2 m
Side slope	Z	1.5
Bottom slope	S_0	0.000045
Manning coefficient	n	0.012
Upstream reservoir depth	y_{UR}	9.163 m
Downstream reservoir depth	y_{DR}	8.837 m
Pool length	L	7254 m
Gate structure width	B_G	24.4 m
Orifice coefficient	C_D	0.75

The time histories of the upstream and downstream depths in a single pool are shown in Fig. 9.

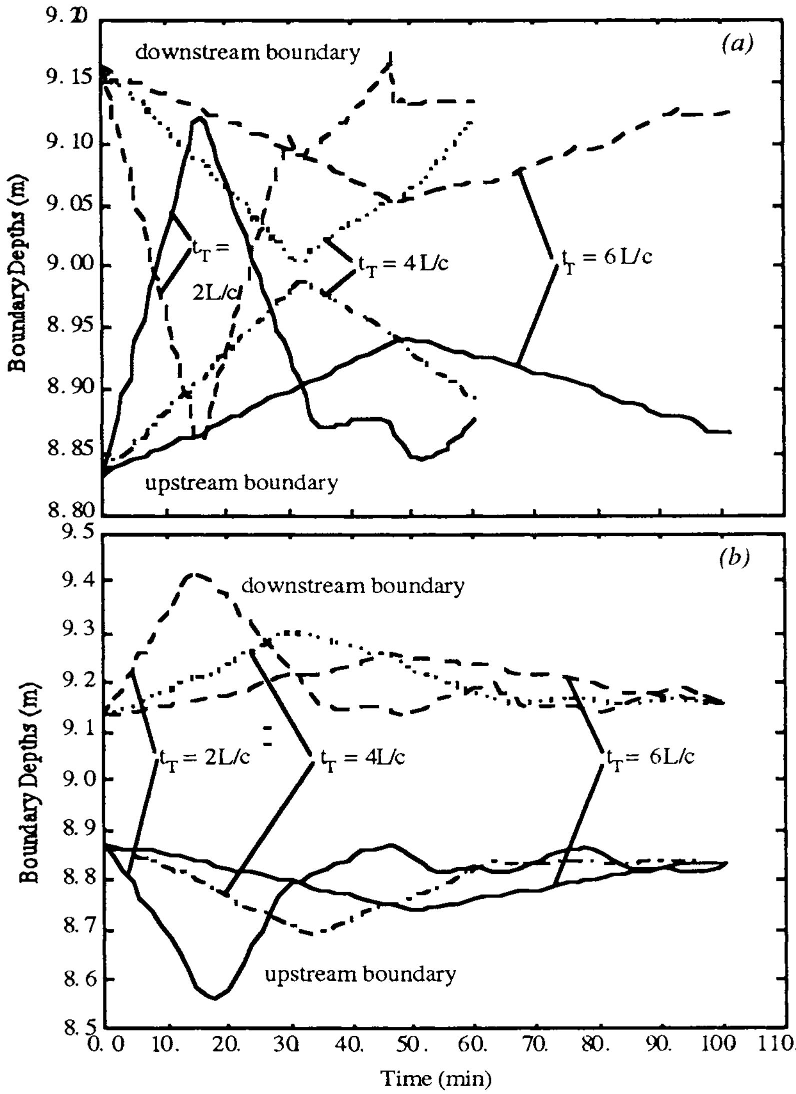


Figure 9 Simulations of nonlinear transients when constant drawdown rate magnitude boundary conditions (based on a wave equation model) are applied. (a) Boundary depths for flow initiation, 0 - 180 m^3/s. (b) Boundary depths for flow arrest, 180 - 0 m^3/s.

Notice the nearly constant drawdown rate magnitudes in all cases. The CSWP drawdown constraints are only satisfied in the slowest (t_T = 6L/c) flow transition. Also, the fastest flow initiation would not be realizable in a series of pools since the depths adjacent to internal gates are required to overlap. Boundary discharge time histories are shown in Fig. 10.

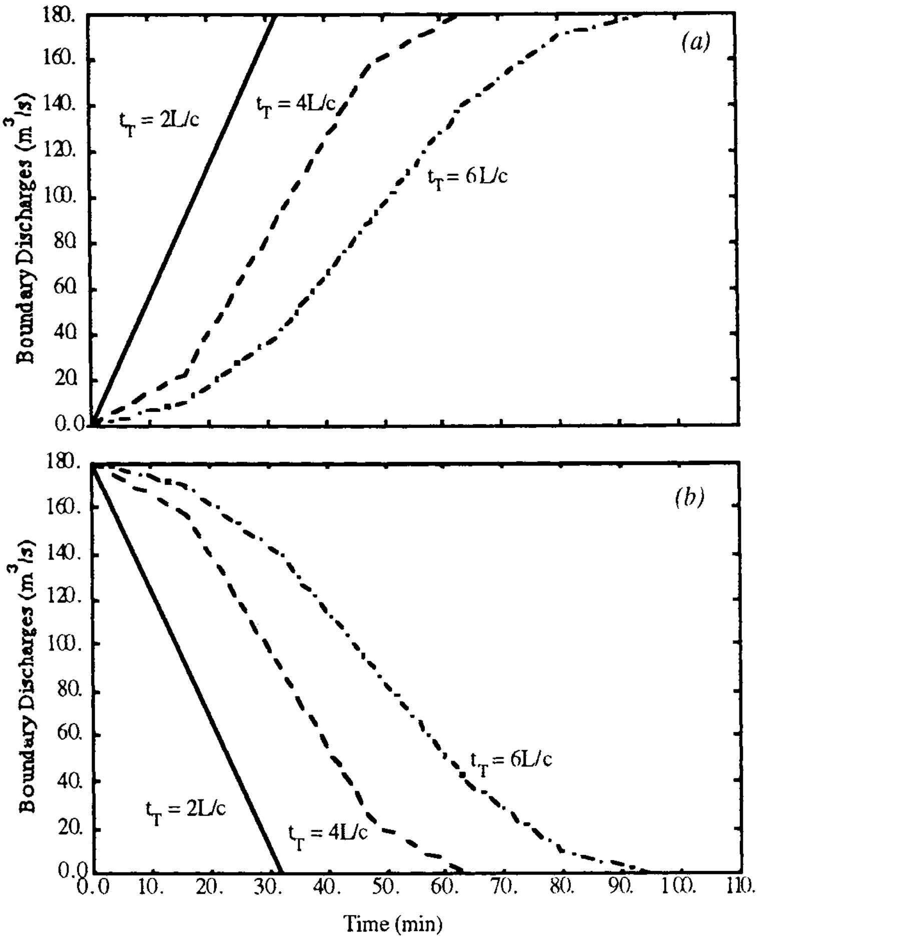


Figure 10 Constant drawdown rate magnitude boundary discharges (based on a wave equation model) for (a) Flow initiation, 0 - 180 m^3/s. and (b) Flow arrest, 180 - 0 m^3/s.

The maximum deviations from the initial values are plotted in Fig. 11 along with the deviations predicted by the wave equation.

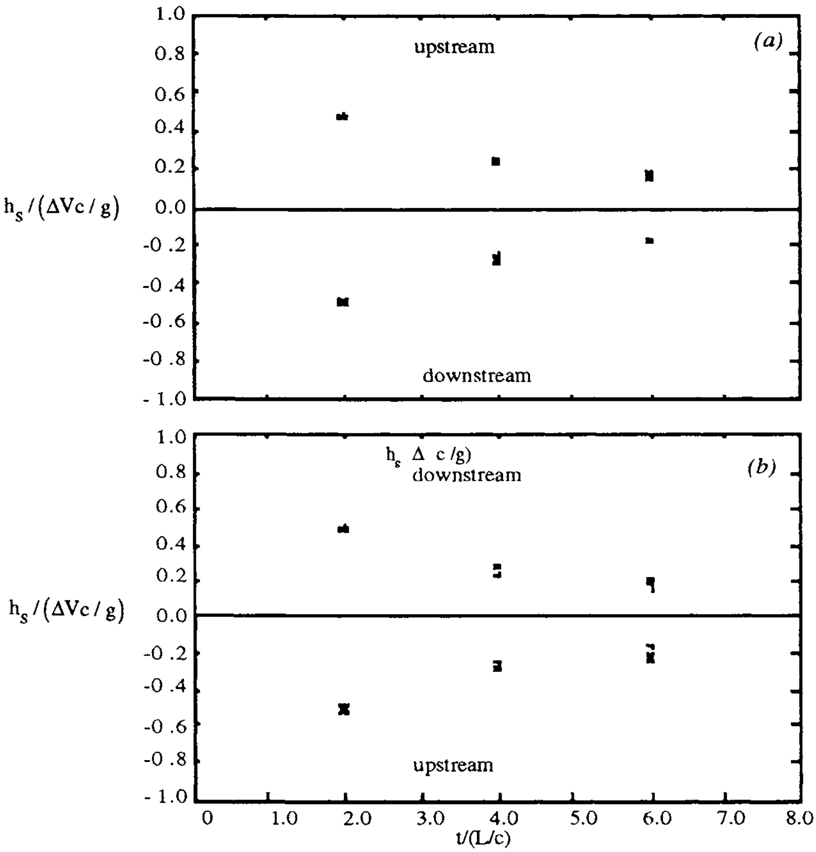


Figure 11 Comparison of maximum drawdown predicted by wave equation (+) and observed in nonlinear simulations (x). (a) Flow initiation, 0 - 180 m^3/s. (b) Flow arrest, 180 - 0 m^3/s.

The agreement is quite good; surge heights are only slightly under estimated for flow arrest. One source of error in the drawdown predicted by the wave equation is neglected water level changes due to the friction slope. In the case of flow arrest, drawdown caused by the friction slope and the dynamic surge add up.

IV. FINITE TIME CONTROL

A. LQR FORMULATION

The original motivation for studying the wave equation was to use the responses derived above as a guide in selecting the penalty matrices for the finite time linear quadratic regulator problem. The discrete-time linear time-invariant formulation of the optimal quadratic control problem requires the minimization of the following scalar cost function [19].

$$J = \sum_{k=0}^{k_f-1} [\mathbf{x}_{k+1}^t \mathbf{Q} \mathbf{x}_{k+1} + \mathbf{u}_{k+1}^t \mathbf{P} \mathbf{u}_{k+1}] + \mathbf{x}_{k_f}^t \mathbf{F} \mathbf{x}_{k_f} \tag{29}$$

subject to the dynamic constraints of the linear model

$$\mathbf{x}_{k+1} = \Phi \mathbf{x}_k + \Gamma \mathbf{u}_k \tag{30}$$

where $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{F}$ are penalty matrices which must be specified. It is well known that the control which minimizes (29) is a time-varying feedback law

$$\mathbf{u}_k = -\mathbf{K}_k \mathbf{x}_k \tag{31}$$

Notice that the state, $\mathbf{x}_k$, and the control, input, $\mathbf{u}_k$, are both deviations from some equilibrium about which the plant model has been linearized. This formulation is useful only when the terminal state is steady--the situation encountered in any imaginable canal operation.

It is not immediately apparent how to solve a constrained optimization problem, such as we have in canal operations, using the linear quadratic regulator formulation. For a given transition time, the drawdown constraints must be satisfied while keeping the terminal state within some reasonable amount of the final steady flow condition. If the elements of $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{F}$ are thought of as Lagrange multipliers, the values which produce the proper constraints can be determined by iteration [19]. The method is unwieldy if each element is dealt with individually. However, with some judgment and insight into the dynamics of the system, the technique is workable.

B. PROCEDURE FOR SELECTING THE PENALTY MATRICES

A common method for choosing the penalty matrices $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{F}$ is "Bryson's rule." The rule involves setting the diagonal elements of $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{F}$ to the inverse square of the maximum deviation of the associated control or state variable. The off-diagonal elements are zero unless there is some physically meaningful reason to couple the state and/or control penalties. Diagonal matrices are used here since we can't think of any practical reason for off-diagonal weighting. The resulting penalty function is the weighted sum of the squares of the state and control deviation at every sampling time

$$J = \sum_{k=0}^{k_f - 1} \left[\sum_{i=1}^{n} \frac{x_{i,k}^2}{x_{i\,max}^2} + \sum_{j=1}^{r} \frac{u_{j,k}^2}{u_{j\,max}^2} \right] + \sum_{i=1}^{n} \frac{x_{i,kf}^2}{x_{i,kf\,max}^2} \tag{32}$$

The results of the wave equation analysis will be used to implement Bryson's rule in the following manner. The transition is defined by specifying t_T and ΔV. The maximum velocity deviation, V_{max}, is just ΔV, and the maximum depth deviation, y_{max} (from Eq. (22)), is $(\Delta VL)/(2gt_f)$. The maximum gate (discharge) deviation, $b_{i\,max}$, is just the difference between the initial and final steady gate position (discharge). The maximum gate velocity (rate of discharge change), $u_{i\,max}$, may be determined from a physical limit or to maintain the gradually varied flow assumption; a nominal value of $b_{i\,max}/t_f$ is chosen. The values of the diagonal elements of **Q** and **P** thus depend on the corresponding state variables as shown in Table 2.

Table 2 Penalty Matrix Formulation

STATE VARIABLE	PENALTY	PENALTY
Depth	**Q**	$(\rho_{Q^y}(2gt_T)/(\Delta VL))^2$
Velocity	**Q**	$(\rho_{Q^v}/\Delta V)^2$
Gate Position	**Q**	$(\rho_{Q^b}/(b_f - b_i))^2$
Gate Discharge*	**Q**	$(\rho_{Q^b}/\Delta Q)^2$
Gate Velocity	**P**	$(\rho_P t_T/(b_f - b_i))^2$
Rate of Change of Gate Discharge*	**P**	$(\rho_P t_T/\Delta Q)^2$

* Alternate Model (See Section II.A.)

The terminal penalty **F** is just

$$\mathbf{F} = \rho_F^2 (t_T/\Delta t)^2 \mathbf{Q} \tag{33}$$

The factor $(t_T/\Delta t)^2$ is added to insure that the same relative terminal penalty is used regardless of the discretization level. The extra factors, $\rho_Q{}^y$, $\rho_Q{}^V$, $\rho_Q{}^b$, ρ_P, and ρ_F, provide the flexibility to shape the response. For example, if $\rho_Q{}^y$ is 1/2, the maximum depth deviation is correct for the constant drawdown rate magnitude transition Eq. (27).

Notice that with the proposed method only four degrees of freedom exist in specifying the cost function (since dividing the cost function by a scalar will not change the minimal solution). Also, it should be pointed out that for a multi-pool system, L may be chosen as the largest pool length or the average if the lengths are nearly the same.

C. FINITE TIME LQR RESULTS

Computational results, which demonstrate the possibilities for shaping the response by changing the "ρ factors," are shown in Fig. 12.

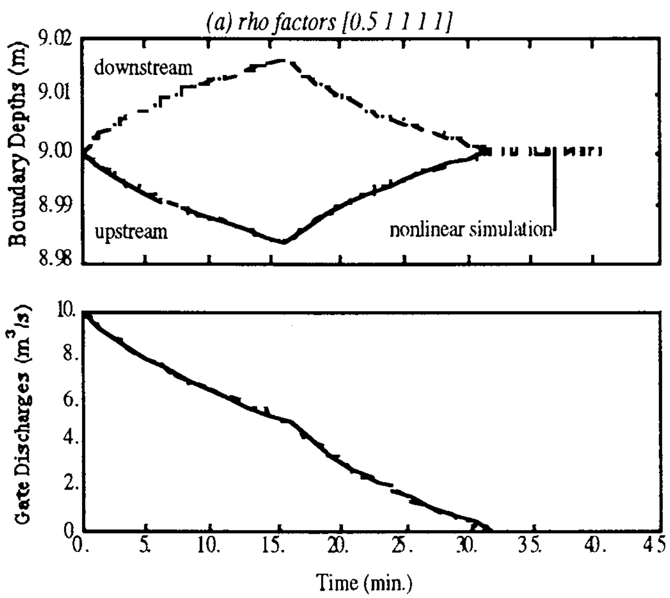
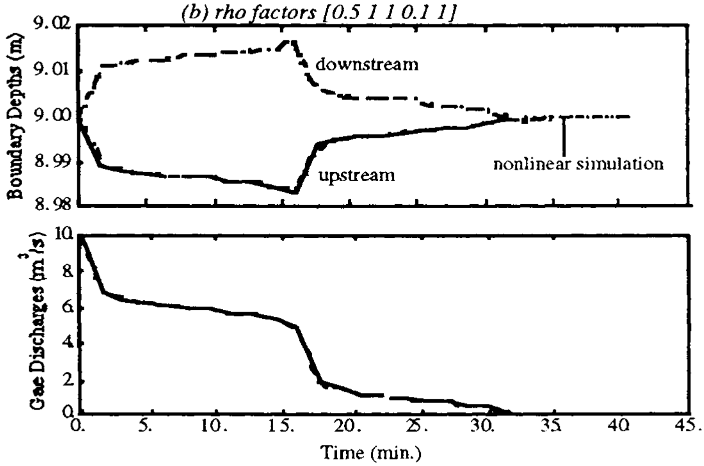
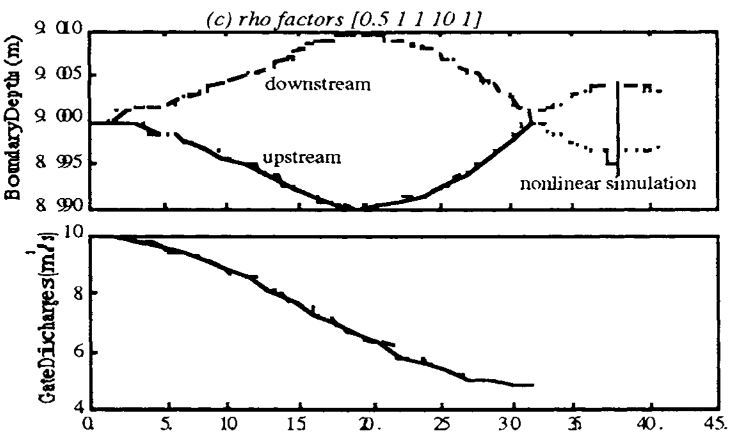


Figure 12 Optimal boundary conditions computed using finite time control algorithm. "ρ factors" $\left(\left[\rho_Q y \; \rho_Q V \; \rho_Q b \; \rho_p \; \rho_F\right]\right)$:
(a) [0.5 1 1 1 1]

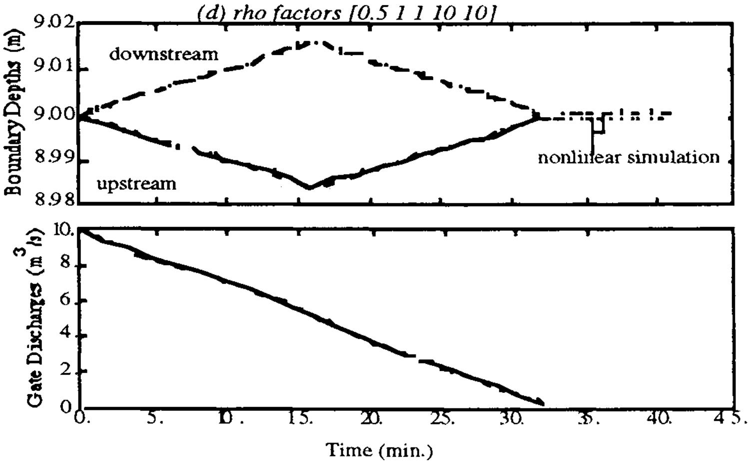


(12b)

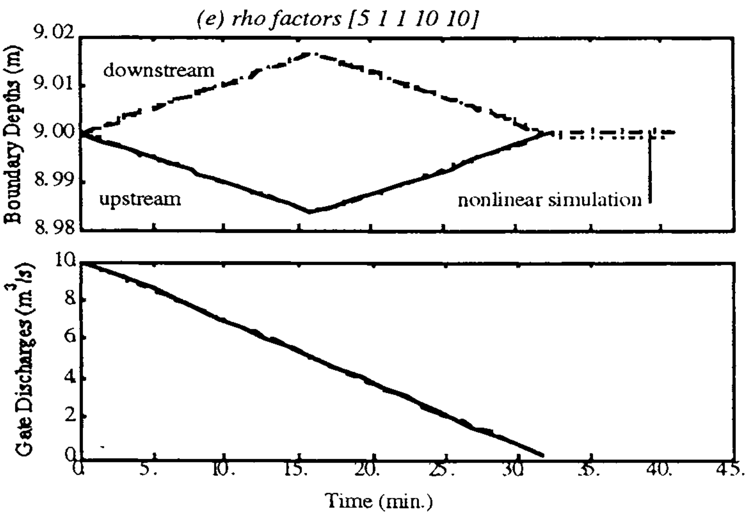


(12c)

Figure 12 Optimal boundary conditions computed using finite time control algorithm. "ρ factors" $\left(\left[\rho_Q y\ \rho_Q V\ \rho_Q b\ \rho_p\ \rho_F\right]\right)$:
(b) [0.5 1 1 0.1 1] (c) [0.5 1 1 10 1]

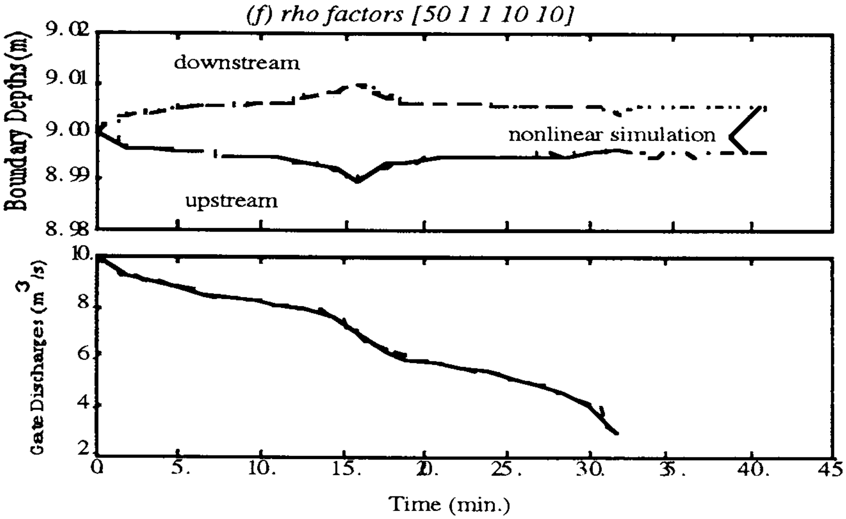


(12d)

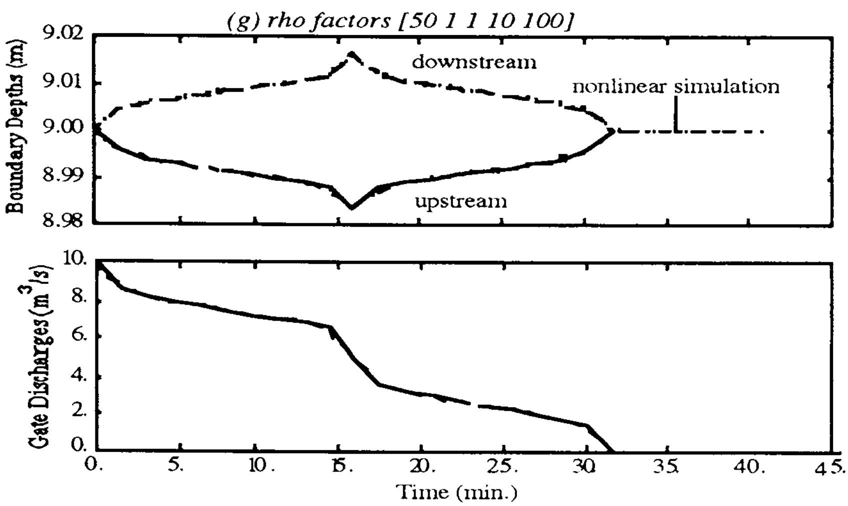


(12e)

Figure 12 Optimal boundary conditions computed using finite time control algorithm. "ρ factors" $\left(\left[\rho_Q y\ \rho_Q V\ \rho_Q b\ \rho_p\ \rho_F\right]\right)$
(d) [0.5 1 1 10 10] (e) [5 1 1 10 10]



(12f)



(12g)

Figure 12 Optimal boundary conditions computed using finite time control algorithm. "ρ factors" $\left(\left[\rho_Q y\ \rho_Q V\ \rho_Q b\ \rho_p\ \rho_F\right]\right)$
(f) [50 1 1 10 10], (g) [50 1 1 10 100]

The gate discharges and the depths at the upstream and downstream boundaries for both linear and nonlinear simulations are displayed. Optimal gate discharges were computed for flow arrest from 10 m^3/s for a single pool using a 10 reach gate discharge input model. Canal parameters are those listed in Table 1 except that the bottom slope is set to zero. The transition time is 31.8 min. (=2L/c).

Studying Fig. 12 we can see how the optimal responses are affected by the choice of the penalty matrices. The responses in Figs. 12a 12d and 12e resemble minimum drawdown rate responses and 12b resembles the minimum time response. Fig. 12c shows the effect of too much gate discharge rate penalty vs terminal penalty. Fig. 12f shows the effect of a heavy penalty on depths. The response resembles one of the constant surge height family responses with the discharge at the boundary being varied in a step-like fashion. These results indicate that even with the normalization of penalty coefficients using known wave equation responses, shaping the response to fit a general set of drawdown constraints can require many trial and error computations. One problem is that it is not possible to penalize and thus control drawdown rate because it is not a state variable.

Summarizing, we have what must be a typical dilemna in developing finite time control schemes for any dynamic system. With an inverse dynamics approach, such as gate stroking, we can force some aspect of the solution to be what is desired, but other aspects of the solution may suffer, that is we may not be making the best trade-off. On the other hand, when linear quadratic optimal control is used, we lose the precise control of specific responses which can make satisfying a hard constraint a tedious trial and error process. Also, if a solution which satisfies the hard constraint is found, it still may not be the best trade-off for other system responses. Perhaps a more sophisticated formulation of the optimal control problem could resolve this dilemma.

V. REGULATION

Closed-loop regulation can benefit several aspects of modern canal operations--e.g. the depth levels at overflow weirs may be regulated to provide the desired outflows in the face of disturbances, unknown initial conditions, or parameter uncertainty. It is recognized that the important design considerations in the regulation problem are disturbance, plant parameter, and sensor noise insensitivity and that the LQR formulation of the regulator problem does not explicitly account for plant parameter insensitivity. Despite these practically important shortcomings, the LQR technique will be employed here. The justification is that we are more concerned here with learning about canal control in general, rather than in designing a specific control system. In fact, we believe that in many cases plant parameter uncertainty will be manageable. Parameters such as channel geometry are straightforward to determine. The tremendous amount of data collected routinely in the operation of a modern canal can likely be used to estimate the less straightforward parameters--such as the gate coefficients and Manning coefficients--with good precision. Our purpose here is to demonstrate the feasibility of LQR based compensator design and to investigate the limited performance achievable with this system due to the inherent time delays.

A. STEADY STATE FEEDBACK GAINS

In the steady state regulation problem, the performance index to be minimized is

$$J = \lim_{k_f \to \infty} \left\{ \sum_{k=0}^{k_f} [\mathbf{x}_{k+1}^t \mathbf{Q} \mathbf{x}_{k+1} + \mathbf{u}_{k+1}^t \mathbf{P} \mathbf{u}_{k+1}] \right\} \tag{34}$$

The control law resulting from the minimization of (34) is

$$\mathbf{u}_k = -\mathbf{P}^{-1} \Gamma^t [\mathbf{R}^{-1} + \Gamma \mathbf{P}^{-1} \Gamma^t]^{-1} \Phi \mathbf{x}_k = -\mathbf{K} \mathbf{x}_k \tag{35}$$

state variable feedback with constant gain matrix **K**, where **R** is the well known steady state solution of a Riccati equation. This problem and existence and uniqueness conditions for its solution are well documented e.g. [19]. The penalty matrices **Q**, and **P** may be scaled using the wave equation responses as was done for the finite time application.

B. STATE ESTIMATION

The feedback scheme discussed thus far is undesirable since it requires measurement of the entire state vector. In many canals, e.g. California Aqueduct, the water levels adjacent to the gates are available [4]. The expense of constructing and maintaining facilities to measure depths and flows throughout the canal system is prohibitive. However, a regulation scheme which makes use of the available measurements can be developed using state estimation.

The general form of a full-order state estimator is

$$\mathbf{z}_{k+1} = \mathbf{E} \mathbf{z}_k + \mathbf{H} \mathbf{u}_k + \mathbf{L} \mathbf{y}_k \tag{36}$$

where the vectors **z**, **u**, and **y** = **Cx**, represent state estimates, control inputs, and measurements respectively. When

$$\mathbf{E} = \Phi - \mathbf{LC} \; ; \quad \mathbf{H} = \Gamma \tag{37}$$

the reconstruction error is

$$\mathbf{e}_{k+1} = (\Phi - \mathbf{LC}) \mathbf{e}_k \tag{38}$$

The decay of the reconstruction error depends upon the eigenvalues of the matrix Φ - **LC**. A well known result is that these eigenvalues may be selected arbitrarily if the system is reconstructible, or equivalently if

$$\mathrm{rank}\{[(\Phi^t)^{n-1}C^t \mid (\Phi^t)^{n-2}C^t \mid \cdots \mid C^t]\} = n \tag{39}$$

where n is the order of the system. It is easily verified, using a wave equation model, that the state in a pool is always reconstructible from depth measurements at the upstream and downstream boundaries. The minimum time estimator gains are the first 2p columns of

$$\Phi^m \begin{bmatrix} C\Phi^{m-1} \\ C\Phi^{m-2} \\ \cdot \\ \cdot \\ C \end{bmatrix}^{-1} \tag{40}$$

where m = n/(2p).

An optimal estimator may be derived by minimizing a scalar performance index analogous to (34). In this stochastic approach to estimator design the weighting matrices which form the penalty function are determined from a detailed knowledge of the disturbances affecting the system and the sensor noise characteristics. Thus the desire for rapid error decay is balanced by the conflicting desire for performance insensitivity to sensor noise [19]. Accurately determining the characteristics of the disturbances and sensor noise is beyond the scope of the present work. In the numerical example which follows, the minimum time gains (40) are used.

C. DECOUPLED STATE RECONSTRUCTION

It can be shown that the state can be reconstructed in each pool of a canal independently. That is, estimates of the states in a particular pool only need be modified using measurements from that pool. Estimator gain computations using a single model of the entire system have produced this independent pool structure consistently. In fact, the "pool decoupled" structure was also reported by Balogun [3] who noted that elements of **L** associated with states and measurements from different pools were much smaller than those associated with states and measurements from the same pool.

The reason for this very regular structure of **L** is clear. Referring to Fig. 5b it is evident that if the boundary depths and control inputs (either the boundary discharges or the gate positions) are known, then the complete boundary conditions (y, and V) can be determined. The complete state of the system for t > L/c may then be determined without knowing the initial condition using the characteristic relations and the values of the depth and velocity variables at the boundaries alone. The state of the system is thus "reconstructed." For any number of pools, knowledge of the boundary depths and control inputs can be used to develop the complete boundary conditions; to reconstruct the state of a pool for $t > L/c$, only the boundary conditions for that pool are needed. State reconstruction is thus decoupled. The design and implementation of the estimator can thus be substantially simplified by treating each pool separately.

Unlike the minimum time to control the state of the system, the minimum time to reconstruct the state of the system (L/c using a wave equation model) is unaffected by the number of pools. Thus, under ideal conditions (when the plant model is exact, no disturbances act, and it is possible to use the minimum time control law and the minimum time estimator gains without causing saturation), the minimum time to bring a system of p pools with r control gates to equilibrium is limited by

$$t_{min} \geq \left(\frac{2p}{r} + 1\right)\frac{L}{c} \tag{41}$$

In such an ideal situation, the only uncertainty is in the initial state of the system.

D. THE COMPENSATOR STRUCTURE AND CONTROL REFERENCES

Since the gate movement dynamics are completely decoupled from the rest of the system and since the gate positions are known, it should not be necessary to include the gate positions as states in the state estimator model.

The state estimator equation for the nonstandard model (10) is

$$\mathbf{z}_{k+1} = \mathbf{E}\mathbf{z}_k + \mathbf{H}\mathbf{b}_{k+1} + \mathbf{L}\mathbf{y}_k \tag{42}$$

where $\mathbf{b}_k$ is the vector of gate positions. Now

$$\mathbf{b}_{k+1} = \mathbf{b}_k + \Delta t\, \mathbf{u}_k \tag{43}$$

and the vector of gate velocities, $\mathbf{u}_k$, is determined from the estimated state, including the gate positions. Partitioning $\mathbf{K}$, we have

$$\mathbf{u}_k = -\mathbf{K}_z\mathbf{z}_k - \mathbf{K}_b\mathbf{b}_k \tag{44}$$

and upon substituting (43) and (44) into (42)

$$\mathbf{z}_{k+1} = (\mathbf{E} - \Delta t\mathbf{H}\mathbf{K}_z)\mathbf{z}_k + \mathbf{H}(\mathbf{I} - \Delta t\mathbf{K}_b)\mathbf{b}_k + \mathbf{L}\mathbf{C}\mathbf{x}_k \tag{45}$$

The remaining closed-loop system equations are

$$\mathbf{x}_{k+1} = \Phi\mathbf{x}_k + \Gamma(\mathbf{I} - \Delta t\mathbf{K}_b)\mathbf{b}_k - \Delta t\Gamma\mathbf{K}_z\mathbf{z}_k \tag{46}$$

$$\mathbf{b}_{k+1} = (\mathbf{I} - \Delta t\mathbf{K}_b)\mathbf{b}_k - \Delta t\mathbf{K}_z\mathbf{z}_k \tag{47}$$

A schematic of the proposed scheme is shown in Fig. 13.

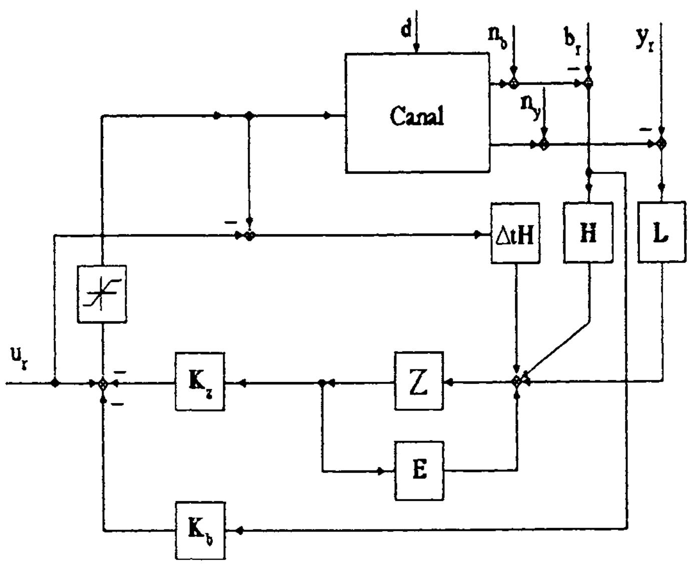


Figure 13 The compensator structure. n_b--gate position sensor noise, n_y--water depth sensor noise, d--disturbances (e. g. wind, rain) which affect the state.

The compensation scheme discussed thus far is set point regulation. The values of the reference variables are held constant at some desired equilibrium state for long periods of time. For flow transitions in which

$$\frac{c|\Delta V|}{2g} \approx \frac{|\Delta Q|}{2\sqrt{gAT}} < h_{max} \tag{48}$$

acceptable performance may be possible with the following regulation technique. A set point may be computed at the desired equilibrium flow, Q_f, along with the associated measurement and feedback gains, **L** and **K** respectively.

It is easily shown that if any nominal trajectory which satisfies the state equations is used as a reference input, the compensation scheme discussed is still valid. That is, the dynamic characteristics of the tracking error are preserved (if the system model is exact). A very simple feedback control scheme would result if a fixed compensator could be used with a reference input for large flow transitions (when Eq. (48) is not satisfied). The feasibility of using one set of values for **E**, **H**, **K**, and **L** over the entire operating range is investigated next.

E. TESTING FOR ROBUSTNESS

The procedure used to design the compensator assumes a known fixed linear plant. We have shown that some aspects of the flow behavior in the canal can be predicted well by a linear model (Fig. 11). However, we know that the energy losses due to friction and momentum convection and the gate control equations are not linear. In evaluating a compensator design, it will be necessary to insure that adequate insensitivity is achieved with respect to both nonlinearities and uncertain and/or changing parameters. The most pressing robustness concern presently is whether the fixed linear finite-order compensator can adequately control a nonlinear distributed plant.

It is well known that the linear optimal quadratic control with state variable feedback is robust according to classical design procedures. Equally well known is the fact that the robustness may deteriorate when the control is determined by using a state estimator [20]. Our concern here is whether or not the compensator will be acceptably insensitive to parameter variations caused by the plant nonlinearity. One approach used to investigate the sensitivity is to study the variation in the closed-loop eigenvalues when Φ and Γ change because of variations in the equilibria used for linearization.

Limited robustness verification was performed for a single pool channel. The eigenvalues of a 22 state lumped model linearized about zero flow are shown in Fig. 14a.

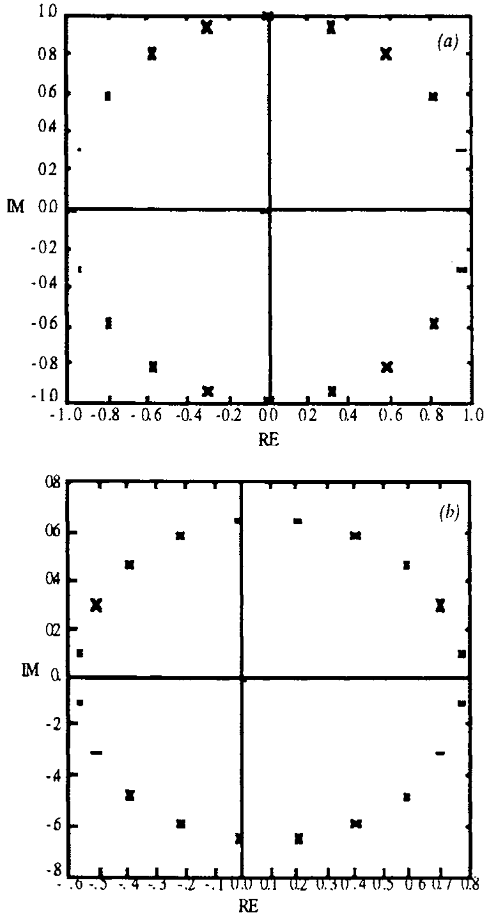


Figure 14 Open-loop eigenvalues of a 10 reach single pool linearized MOC model. (a) Linearized at Q = 0. (b) Linearized at Q = 180 m^3/s.

The fact that the moduli of all of the eigenvalues are nearly one indicates that there is very little damping in the open-loop system. In fact there is no mechanism for dissipation at zero flow and the moduli of all the eigenvalues would be exactly one were it not for the interpolations caused by the nonzero bottom slope. Fig. 14b shows the open-loop eigenvalues of the lumped model linearized about a nominal flow of 180 m^3/s. The reduction in the eigenvalues' moduli indicates a realistic increase in damping caused by the nonzero friction slope. Part of the moduli reduction is an artifact of the interpolations used in the numerical approximation of the characteristic equations. Notice that the moduli of the higher frequency modes are smaller than those of the lower frequency modes.

The estimator gain matrix, **L**, and the feedback control gain matrix, **K**, are computed using the model linearized about zero flow. The minimum time algorithm is used to compute **L**; **K** is from (35). The weighting parameters are: $\rho_Q{}^y = 1.0$, $\rho_Q{}^V = 1.0$, $\rho_Q{}^b = 1.0$, and $\rho_P = 0.01$; $t_T = 15.9$ min.

The eigenvalues of the closed-loop system (Fig. 16a) should be the combined eigenvalues of $(\Phi - \Gamma K)$ (Fig. 15a) and **E** $(\Phi - LC)$ (Fig. 15b) from the separation principle. Notice that the moduli of the "estimator eigenvalues" shown in Fig. 15b are much smaller (but still not zero) than the moduli of the inner ring of eigenvalues in the combined system (Fig. 16a).

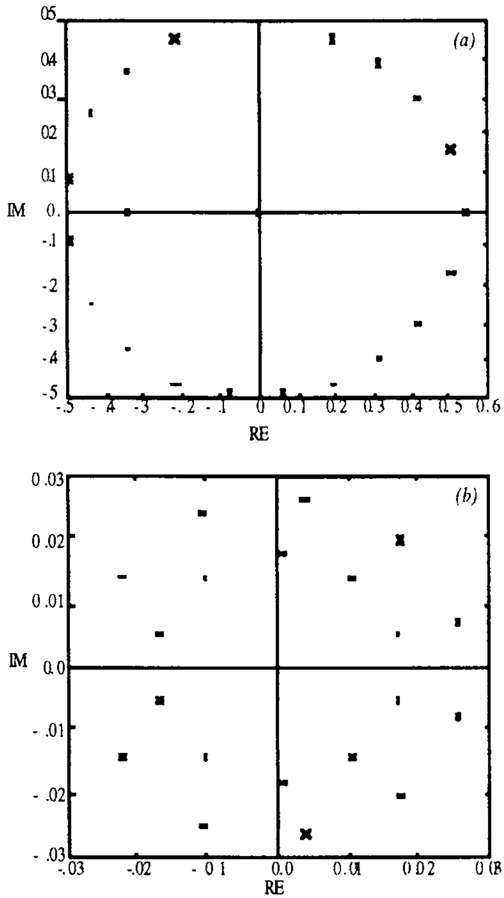


Figure 15 Closed-loop eigenvalues (10 reach single pool model linearized at Q = 0) of (a) $(\Phi - \Gamma K)$, and (b) $(\Phi - LC)$.

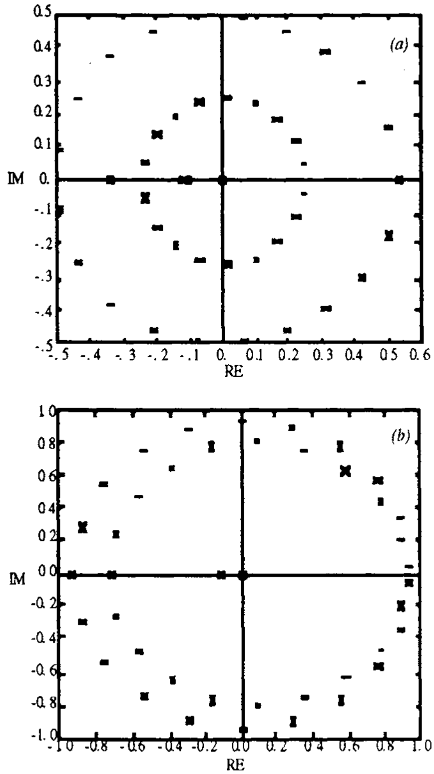


Figure 16 Closed-loop eigenvalues of combined estimator and plant system (10 reaches, one pool). Compensator design based on model linearized at Q = 0. (a) Plant model linearized at Q = 0. (b) Plant model linearized at Q = 180 m^3/s.

Apparently, placing the estimator poles at the origin is a sensitive computation. The discrepancy was found to be due to the numerical precision of **L** alone—double precision in Fig. 15b and single in Fig. 16a. This result is striking and indicates that in practice, where much larger error is certain to occur, some compromise is likely in the realization of the minimum time estimator. Nevertheless, the damping of all the closed-loop poles is still quite satisfactory. The outer ring of closed-loop eigenvalues (Fig. 16a) are the "controller eigenvalues." Their positions appear unchanged from those shown in Fig. 15a.

In Fig. 16b the closed-loop eigenvalues are shown for the case where the system model (Φ and Γ) is linearized about full flow (180 m^3/s). The controller and estimator gain matrices, **K** and **L**, and the estimator state equation matrices **E** and **H**, however, are unchanged from their nominal design values. Notice the dramatic change in the eigenvalue locations. Although all of the eigenvalues are still stable, there is much less damping . In fact, the damping in many modes appears to be less than that in the open-loop system at 180 m^3/s (Fig. 14b)!

This really is not so surprising since we have used a gate input model to develop **K**, **L**, and **E**. The single linearized approximation is inaccurate over the entire flow range of the CA. A gate input model is used because the gate position, not the discharge, is controllable in actual operations. Some preliminary work indicates that it would in fact be feasible to use a gate discharge model for real-time control. The control law would specify a gate discharge. Estimations of the state could then be used to move the gate position so that the desired discharge would be realized. This is not a completely standard use of linear quadratic regulation. Although not pursued, it appears that incorporating the nearly linear discharge equations in such a modified scheme might reduce the deterioration seen in Fig. 16b.

This approach to robustness verification rapidly becomes unwieldy for higher order multiple pool models. Also, it is difficult to check the robustness of the closed-loop system when the canal model order is higher than the compensator order. An equally valid (and perhaps easier to interpret) procedure for testing the robustness of the discrete linear compensator is of course to test its performance by simulation using a nonlinear flow model for the fluid dynamics at various discretization levels.

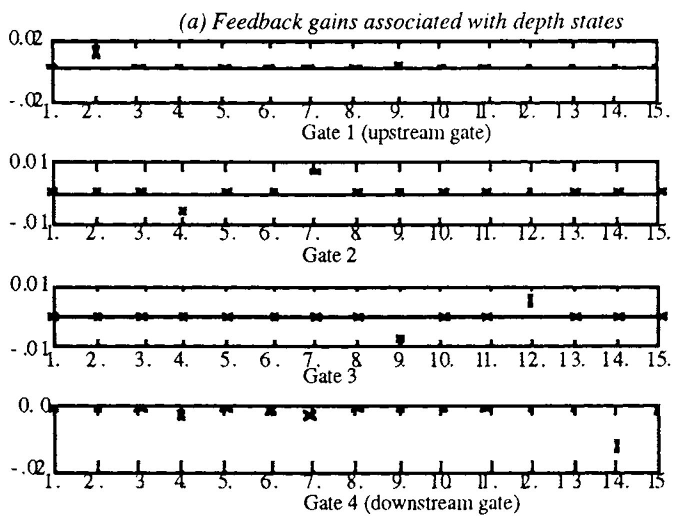
VI. EXAMPLE APPLICATION

An example application of the discrete time LQR controller should help to clarify some practical application details and demonstrate the controller performance in a realistic situation. Example canal parameters are close to those of the California Aqueduct (see Table 1). The situation we shall study is flow arrest from 180 m^3/s. The shutdown situation is more difficult to control than startup since the residual transient reflects almost perfectly from the closed gates. Also, when regulation is used, the gate cannot be freely moved in the neighborhood of its final rest position. A three pool model is chosen to reduce the computational burden but there is no reason why results would differ substantially if more pools were used. Every possible pool boundary condition is represented—upstream reservoir, downstream reservoir, and no reservoir.

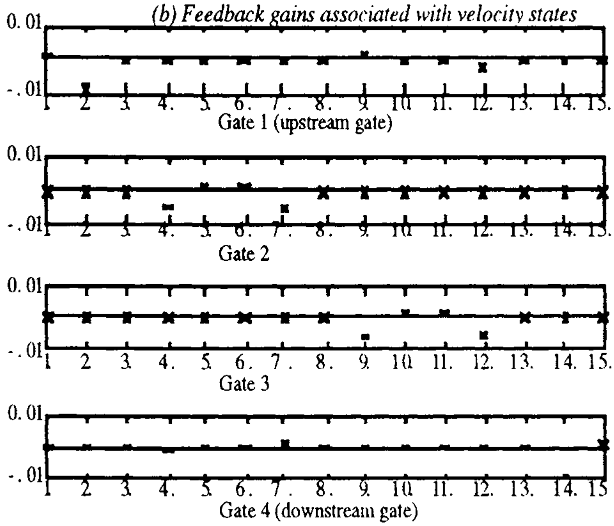
A 34 state model (4 reaches per pool - 15 depth states, 15 velocity states, and 4 gate position states) linearized about zero flow was used to compute the controller feedback gains, **K**. The penalty coefficients chosen were $\rho_Q{}^y$ = 1.0, $\rho_Q{}^V$ = 1.0, $\rho_Q{}^b$ = 1.0, ρ_P = 0.01 (t_T = 15.9 min.). With the gate position states truncated (resulting in a 30 state model), the minimum time estimation algorithm was used to compute the state estimator gains, **L**.

Fig. 17 shows the feedback gains associated with depth (Fig. 17a) and velocity (Fig. 17b) states, and illustrates the magnitude of the feedback gains controlling each gate. They are tabulated separately for the 15 depth and velocity states throughout the entire canal. The gate locations are upstream of the first depth (and velocity) state, between states "5" and "6" and "10" and "11", and downstream of the last state. Notice that the states next to those adjacent to a particular gate, dominate the control law for that gate—e.g. depth and velocity "2" for gate "1". The magnitude of gains for the upstream and downstream gates is about twice as large as for the internal gates—but twice as many states affect the internal gates (two depth and two velocity).

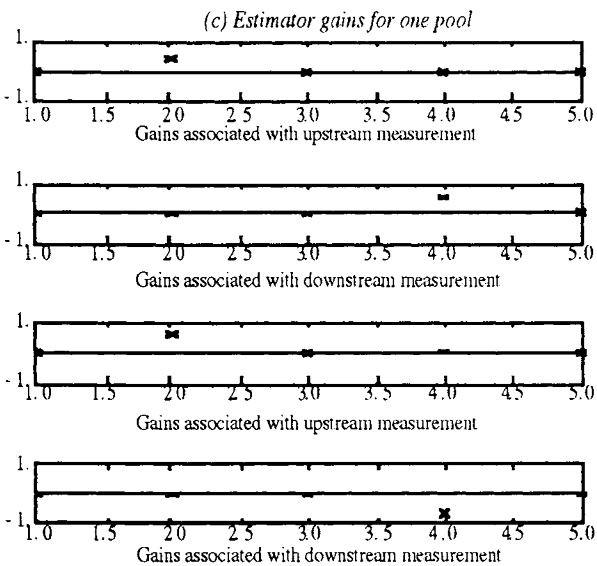
Figure 17 Compensator gains. The columns of the controller gain matrix, **K**, are partitioned according to their associated state variable type. Each row, which corresponds to the feedback gains controlling a particular gate, is graphed separately. (a) Depth feedback gains. (b) Velocity feedback gains. (c) Estimator gains. Only the nonzero elements of rows of the estimator gain matrix, L, associated with states in a single pool are displayed. These rows are further partitioned according to their associated state variable type (depth and velocity). Columns, which correspond to measurements from the upstream and downstream ends of a pool are displayed separately.



(17a)



(17b)



(17c)

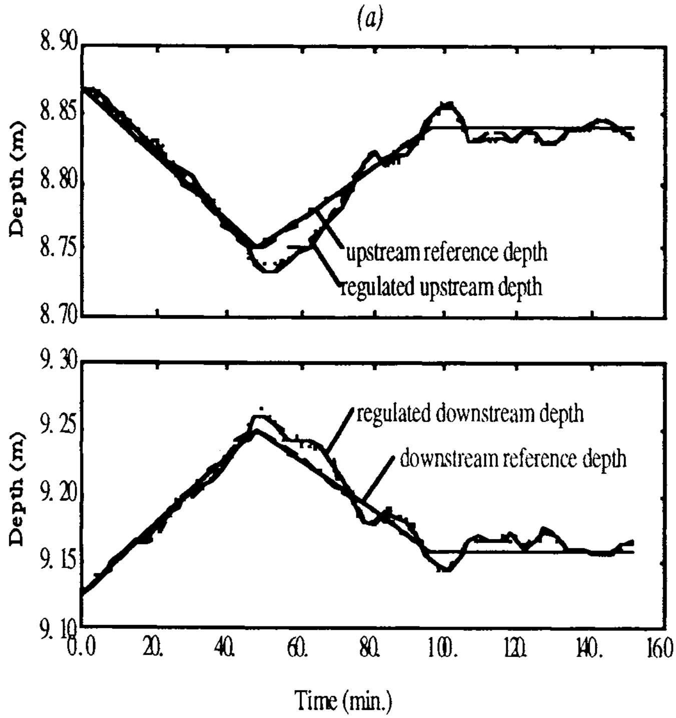
Top pair represents depth gains, bottom pair, velocity gains.

Note how the gates respond to disturbances. All depth feedback coefficients downstream of a gate are positive; the coefficients upstream of a gate are negative. All velocity feedback coefficients are negative. A positive surge traveling upstream would accompany a reduction in velocity. The perturbations multiplied by the feedback coefficients are both positive. But since the feedback is negative, such a surge would result in the gate being closed. A positive surge traveling downstream similarly would result in the gate downstream of the surge being opened. Thus, the feedback seeks to anticipate the arrival of the surge at the gate, and adjust the gate position to accommodate or pass the disturbance (as opposed to reflecting it).

A feedback gain structure in which the states in pools adjacent to a gate dominate the control of that gate was also reported by Balogun [3]. This pattern of feedback gains does not decouple the pools in the way that the minimum time estimator gains decouple state reconstruction. The states in a particular pool influence mainly the gates which bound that pool but adjacent pools also influence these gates. Although, it hasn't been proven, it seems like the near zero gains are unimportant and perhaps even artifacts of modeling and computational errors—especially in light of the physical interpretation of the gains above. Perhaps little or nothing would be lost in setting them to zero.

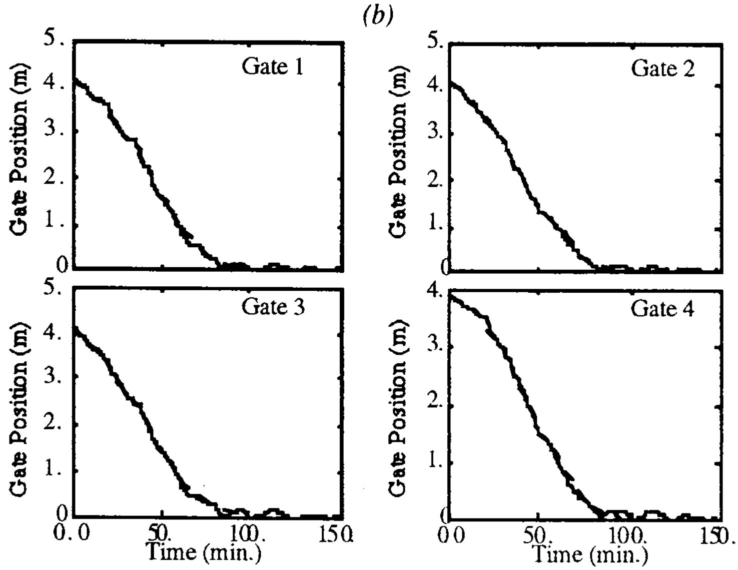
The minimum time estimator gains were found to be identical for each pool so gains for only one pool are shown (Fig. 17c). The graphs show how the upstream and downstream depth measurements are used to modify the estimated depth and flow states. The measured information is used predominantly to modify the estimate of the states next to those adjacent to the gates. There is also an obvious physical explanation for the estimator gain structure. The measurements at the gates essentially determine the boundary condition. Since this boundary condition will propagate to the states adjacent to the boundary states in the next time interval, it is this adjacent state which the estimator continues to adjust to refine the estimation of the complete state.

The reference transient is one for which the drawdown rate magnitude is constant adjacent to the check gates throughout the flow change—for a wave equation model. In order to avoid initial and final condition mismatch (between the wave equation and St. Venant Eqn.), the constant drawdown rate magnitude depth reference is added to a straight line from the known nonlinear initial and final steady conditions. The gate position references are generated by running a nonlinear MOC simulation in which the flow discharges through the gates are controlled to produce the constant drawdown rate transition. Gate positions are calculated using the specified discharges, and the simulated depth responses. In the simulation presented, a 168 state nonlinear first order MOC model is used to model the flow dynamics. The results were found not to be sensitive to the canal flow model order. Notice (Fig. 18a) how the compensator tries to force the St. Venant Eqn. model to respond like the wave equation with impressive results.

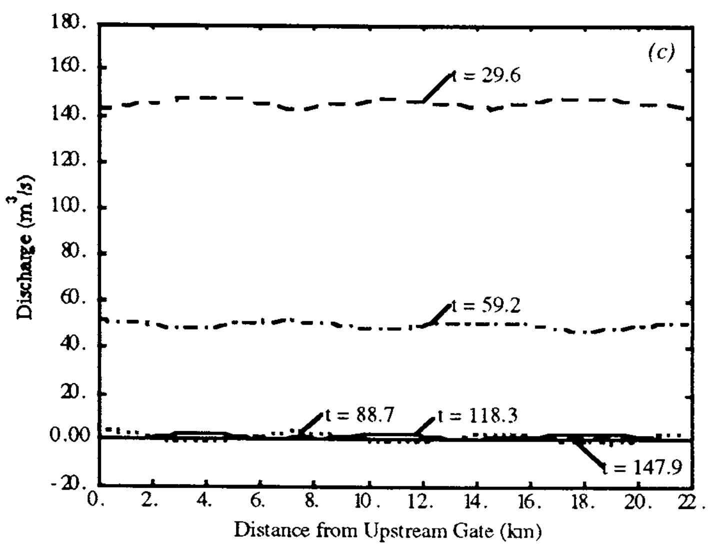


(18a)

Figure 18 Regulated transient (180 - 0 m^3/s flow arrest) with constant drawdown rate magnitude reference. (a) Boundary depths. (b) Gate positions (regulated, solid; reference, dashed). (c) Discharge profile for regulated transient.


(b)
Gate 1
Gate 2
Gate 3
Gate 4
Gate Position (m)
Time (min.)

18(b)


(c)
t = 29.6
t = 59.2
t = 88.7
t = 118.3
t = 147.9
Discharge (m^3/s)
Distance from Upstream Gate (km)

18(c)

Flow is almost completely arrested after 89 min. The flow arrest is primarily due to the accuracy of the reference transient calculated from the wave equation. It may be better to use the depths from the nonlinear MOC simulation with constant drawdown rate discharge conditions as depth references. However, the mismatch between the idealized depth profiles and the simulated response tends to generate some disturbances in this simulation which the regulator seems to damp out albeit slowly. Notice how the gate motions get smaller as time increases—this is a reassuring sign!

VII. CONCLUSIONS

The problem of controlling the flow in a canal system has been addressed in this work using well known methods of modern control theory. Completely satisfactory procedures for scheduling the gate motions optimally and for regulating the flow throughout the canal do not exist. A discrete-time linearized first order model based on the method of characteristics was chosen for regulator design after analyzing and trying several other approaches. Its strengths compared with other lumped models were more accurate handling of boundary conditions, better approximation of the truncated spectrum for a given model order, and less computational burden when used to generate a regulation algorithm.

Attempts to derive an optimal quadratic feedback controller for finite time flow transients were only partially successful. Satisfying realistic hard constraints encountered in canal operations required many iterative trials. However, families of constant surge magnitude and drawdown rate magnitude responses derived from the wave equation proved to be applicable to realistic canal transitions. The wave equation analysis was also useful for selecting the penalty matrices for the (infinite time) optimal quadratic regulator. Very good results have been obtained using a wave equation transient as a reference input in conjunction with an optimal quadratic regulator.

The structure of both the LQR controller gains and estimator gains is degenerate--a few states dominate the gate control algorithm. The gain structure is understandable considering the wave propagation properties of the canal. Substantial simplification of the general LQR scheme appears possible. Perhaps the gain computation scheme could also be simplified.

The possibility of using fixed linear quadratic compensation for a very large flow transition has been demonstrated. A numerical example of a compensator based on a modest 30 state (neglecting gate positions) model regulating a realistic 168 state nonlinear model was presented. The compensator design has thus been shown to be robust in the face of realistic dynamic nonlinearities. Even more important is the success in controlling a system of substantially higher order than the model on which the design was based. We consider this to be the greatest robustness concern when applying lumped control theory to a distributed parameter system.

Nevertheless, the cases studied were special in many ways. Turnout flows were not included, each pool in the canal was of equal length, the Froude number was small, etc. It is felt by the authors that further research in the area would be beneficial for both canal operation and design. The importance of open channel flow dynamics and modeling in guiding controller design and checking results was also recognized by the authors--there is a danger in blindly applying "generalized control methodologies." Further investigation would benefit from the involvement of researchers with open channel flow expertise.

VIII. REFERENCES

1. Shand, M. J., "Final Report--The Hydraulic Filter Level Offset Method for the Feedback Control of Canal Checks," *University of California Hydraulic Engineering Laboratory*, Berkeley, CA (1968).

2. Buyalski, C. P., and Falvey, H. T., "Stability of Automatic Canal Systems Utilizing the Frequency Response Method," *International Association of Hydraulic Research, Symposium on Fluid Motion Stability in Hydraulic Systems with Automatic Regulators*, Romania (1976).

3. Balogun, O. S., "Design of Real-Time Feedback Control for Canal Systems Using Linear Quadratic Regulator Theory," *Ph.D thesis, Department of Mechanical Engineering*, University of California at Davis (1985).

4. Balogun, O. S., Hubbard, M., and DeVries, J. J., "Automatic Control of Canal Flow Using Linear Quadratic Regulator Theory," *Journal of Hydraulic Engineering*, ASCE, Vol. 114, No. 1, pp. 75-102 (1988).

5. Hubbard, M., DeVries, J. J., and Balogun, O. S., "Feedback Control of Open Channel Flow with Guaranteed Stability," *Proceedings of the 22nd Congress of IAHR*, Lausanne (1987).

6. Gomez, M., Rodellar, J., and Quer, J. L., "Real Time Control of Discharge Deliveries in Open Channels," *Hydrocomp*, Dubrovnik, Yugoslavia (1989).

7. Rodellar, J., Gomez, M., and Martin Vide, J. P., "An Algorithm for Real-Time Control of Open Channel Flow," *Hydrosoft*, Vol. 1, No. 4, pp. 176-180 (1988).

8. Papageorgiou, M. and Messmer, A., "Flow Control of a Long River Stretch," *Automatica*, Vol. 25, No. 2, pp. 177-183 (1989).

9. Strelkoff, T., "Numerical Solution of the Saint-Venant Equations," *Journal of the Hydraulics Division*, ASCE, Vol. 95, HY3, pp. 861-875 (1969).

10. Falvey, H. T. and Luning, P. C., "Gate Stroking," *United States Department of the Interior--Bureau of Reclamation*, Paper REC-ERC-79-7, Denver (1979).

11. Robinson, A. C., "A Survey of Optimal Control of Distributed-Parameter Systems," *Automatica*, Vol. 7, No. 3, pp. 371-388 (1971).

12. Tzafestas, S. G. and Stavroulakis, P., "Recent Advances in the Study of Distributed Parameter Systems," *Journal of the Franklin Institute*, Vol. 315, No. 5/6 (1983).

13. Strelkoff, T., "One-Dimensional Equations of Open-Channel Flow," *Journal of the Hydraulics Division*, ASCE, Vol. 96, HY1, pp. 223-251 (1970).

14. Fread, D. L., "Flood Routing: A Synopsis of Past, Present, and Future Capability," *Proceedings of the International Conference on Rainfall-Runoff Modelling*, Miss., May (1981).

15. Wylie, E. B. and Streeter, V. L., Chap. 16, *Fluid Transients*, McGraw-Hill Advanced Book Division, New York, N. Y. (1978).

16. MacLaren, J. F. T., Tramschek, A. B., Sanjines, A., and Pastrana, O. F., "A Comparison of Numerical Solutions of the Unsteady Flow Equations Applied to Reciprocating Compressor Systems," *Journal Mechanical Engineering Science*, Vol. 17, No. 5, pp. 271-279 (1975).

17. Takahashi, Y., Rabins, M. J., and Auslander, D. M., *Control and Dynamic Systems*, Addison-Wesley, Reading, Mass. (1970).

18. Bodley, W. E. and Wylie, E. B., "Control of Transients in Series Channel with Gates," *Journal of the Hydraulics Division*, ASCE, Vol. 104, HY10, pp. 1395-1407 (1978).

19. Kwaakernaak, H. and Sivan, R., *Linear Optimal Control Systems*, Wiley-Interscience, New York, N. Y. (1972).

20. Lehtomaki, N. A., Sandell, Jr., N. R., and Athans, M., "Robustness Results in Linear-Quadratic Gaussian Based Multivariable Control Designs," *IEEE Transactions on Automatic Control*, Vol. AC-26, No. 1, February, pp. 75-93 (1981).

Analysis and Control of Nonlinear Singularly Perturbed Systems under Sampling [1]

J.P. Barbot*, M. Djemai*, S. Monaco** and D. Normand-Cyrot*

*Laboratoire des Signaux et Systèmes, CNRS/ESE,
Plateau de Moulon 91190 Gif sur Yvette, France.
E-mail : barbot@lss.supelec.fr
Fax : (33 1) 69 41 30 60

**Università di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
E-mail : monaco@itcaspur.caspur.it
Fax : (39 6) 44 58 53 67

[1]The first part of this work concerning the discretization of NLSP systems was partially presented in the Ph.D. dissertation of N. Pantalos

I Introduction

In practice, a considerable amount of systems are singularly perturbed. These systems have the property to be separated into a fast and slow dynamics which allows the use of reduction techniques and composite control. Examples of singularly perturbed systems are: robots with flexible joints ([30, 31]), a Belousov-Zhabotinsky reaction ([2]), power systems ([18, 29]), induction motors ([6]), missiles ([10]), etc.

Even if the particular structure of singularly perturbed systems simplifies the analysis and the control design, it does generate some problems related to systems under sampling. Since nowadays it is common to use computers for the control of linear and nonlinear systems, the analysis and the design of singularly perturbed systems under sampling turn out to be an important topic.

Discretization schemes for NonLinear Singularly Perturbed (NLSP) systems have been proposed in [4, 28], exhibiting some of the problems occuring under sampling. Thanks to an adequate choice of the sampling period in relation with the small parameter ε, it is possible to examine the dynamics of the fast and slow components under fast and slow sampling procedures. This is comparable to the change of time scale used in the continuous-time case. Arguing so, it is possible to define a reduced sampled system on the basis of which a slow control scheme can be designed. This requires the knowledge of the full continuous-time singularly perturbed system. However, in practice the continuous-time fast dynamics is quite usually unknown so that one is faced with the problem of designing a digital control scheme on the basis of the slow reduced continuous-time dynamics only. This problem is discussed in the paper.

Firstly, it is shown that the operation of sampling and reduction do not commute when input output behaviours at sampling time are compared. This is mainly due to the appearance of the input in the output function when reduction of the continuous-time dynamics is performed. In fact, as usual under sampling and discussed in [21] in a nonlinear context, undesirable zeroes appear as well as an input delay appears when sampling the reduced system. This leads to unstability.

The problem has been widely investigated in the linear case. Based on the work of F. Esfandiari and H. Khalil ([7]), in which they propose to add a delay in the transfer function of the reduced system, a specific digital control has been proposed in [3] for linear systems with Unmodelled High-Frequency Dynamics (UHFD). Robust stabilization of linear systems with a structured unmodelled high-frequency dynamics has been widely investigated ([3, 7, 13, 14, 16, 27, 34]), using singular perturbation techniques.

In the nonlinear context, on the basis of a specific class of dynamics, we propose a digital control strategy which is delayed with respect to the data acquisitions (the output measurements). Assuming the fast dynamics unknown but stable, this strategy allows the fast dynamics to converge to the quasi steady state solution and satisfies the control objective with stability.

The paper is organized as follows: in the first part, we do some recalls on nonlinear singularly perturbed systems in Section II and we discuss sampled representation of nonlinear singularly perturbed systems under fast and slow sampling in Section III. The second part concerns the control of systems with unmodelled high frequency dynamics on the basis of characteristic features of singular perturbation techniques and the non commutativity of the sampling and reduction procedures. In Section IV, we analyse the non commutativity of Sampling before Reduction (SR) and Reduction before Slow Sampling (RS). In Section V we propose a digital control procedure introducing a delay between the data acquisitions and input sampling. Many examples are discussed in the paper, in particular, two illustrative examples are given in Section VI. Some conclusions end the paper.

II Some Recalls on NLSP Systems

Hereafter, some recalls on the integral manifold approach for continuous time NonLinear Singularly Perturbed (NLSP) systems case are given (see, for example, [2, 8, 12, 16, 24] for further details).
Consider the singularly perturbed nonlinear system Σ_ε:

$$\Sigma_\varepsilon \quad \begin{aligned} \dot{x} &= f(x,z,u) \qquad x(t_0) = x_0 && (1)\\ \varepsilon\dot{z} &= g(x,z,u) \qquad z(t_0) = z_0 && (2)\\ y &= h(x,z,u,\varepsilon) && (3) \end{aligned}$$

where the state vectors x and z belong respectively to $\mathcal{M}_{n_s}$ and $\mathcal{M}_{n_f}$ which are real manifolds of dimensions n_s and n_f, respectively, the control vector u belongs to $\mathcal{R}^p$, and the output vector y to $\mathcal{R}^q$, $\varepsilon > 0$ is the small perturbation parameter. The functions f, g, h and u are assumed to be sufficiently and continuously differentiable in their arguments and bounded as $\varepsilon \to 0^+$.

Denoting by $\mathcal{D}$ the region of interest, the following assumptions are classical

Assumption A. 1 *(Existence condition) The Jacobian* $\left\{\frac{\partial g(x,z,u)}{\partial z}\right\}$ *is nonsingular for all* $x, z, u \in \mathcal{D}$.

Assumption A. 2 *(Attractivity condition) For all* $x, z, u \in \mathcal{D}$, *the real parts of the eigenvalues of* $\frac{\partial g(x,z,u)}{\partial z}$ *are smaller than a fixed negative number* $-\lambda$, *i.e.*

$$Re\left\{\frac{\partial g(x,z,u)}{\partial z}\right\} < -\lambda < 0$$

so that z *and* x *represent the fast and slow dynamics, respectively.*

If the fast dynamics is stable, then, after a small time period, it converges to a manifold, called the integral manifold or slow manifold. This manifold is characterized by the integral manifold condition or invariance condition. Otherwise, one looks for a control law, denoted by u_f, which brings $z(t)$ to the slow manifold. For this reason, the control vector $u(t)$ in (1) and (2), can be decomposed as:

$$u = u_s(x, v(t), \varepsilon) + u_f(x, z, v(t), \varepsilon) \tag{4}$$

where $v(t)$ is an external control vector, and with u_f equal to zero on the slow manifold and u_s computed to stabilize the set control objective. Control of this type (4) is called composite control ([5, 11, 16, 17, 32]).

Let M_ε the n-dimensional integral slow manifold, defined as:

$$M_\varepsilon = \{(x, z) : z = \Phi(x, u_s, \varepsilon)\}$$

characterized from (2) by the invariance condition $g(x, \Phi, u_s) = \varepsilon\dot{\Phi}$ i.e.:

$$g(x, \Phi, u_s) = \varepsilon(\frac{\partial \Phi}{\partial x} + \frac{\partial \Phi}{\partial u_s}\frac{\partial u_s}{\partial x})f(x, \Phi, u_s) + \varepsilon(\frac{\partial \Phi}{\partial u_s}\frac{\partial u_s}{\partial v})\dot{v} \tag{5}$$

where $\dot{v}$ denotes the time derivative of the external input v. Roughly speaking, condition (5) ensures that if the system is initialized at any point on the manifold M_ε then, under the control (4), the state evolution remains on Σ_ε.

Moreover, if ε is close to zero, one can approximate (5) by the quasi-steady state solution $\bar{\Phi}(x, u_s)$, which verifies (5) for $\varepsilon = 0$, i.e.

$$g(x, \bar{\Phi}, u_s) = 0 \tag{6}$$

Remark: Making use of the Gröbner formula [9], the solution $\bar{\Phi}$ can locally be computed as proposed in [2], i.e.

$$\bar{\Phi} = e^{S(.,z_0)}(I_d)_{|z=z_0}$$

with $S(., z_0) = -L_{Dg(x,z_0,u)}$, where the matrix D denotes the inverse of the Jacobian matrix of $g(x, z, u, 0)$ with respect to z evaluated at the initial state $(x^T, z_0^T, u)^T$ (see Appendix 4).

A crucial assumption for ensuring the existence of a solution to (5) and the attractivity of the slow manifold is given by the Tikhonov's theorem [16, 33] as recalled hearafter.

Solving (5) exactly with respect to $\Phi(x, u_s, \varepsilon)$ is, in general, a difficult task. A classical method consists to consider the Taylor series expansion of Φ with respect to ε, setting:

$$\Phi(x, u_s, \varepsilon) = \Phi_0(x, u_s) + \varepsilon\Phi_1(x, u_s, \dot{u}_s) + O(\varepsilon^2) \tag{7}$$

with $\Phi_0 = \bar{\Phi}$.

Remarks:

i) Since ε is close to zero, low order approximations of Φ are oftently sufficient. These systems will be considered in the two last sections, because, in this case, the steady state reduced dynamics remains linear in u, so simplifying the control design.
ii) The functions Φ_i can be obtained by substituting (7) into (5) and equating terms of the same powers of ε. This procedure requires that the functions f and g in (1) and (2) are also expanded as power series in ε.

Substituting Φ to z in (2) the reduced-order slow model is obtained

$$\dot{x} = f(x, \Phi(x, u_s, \varepsilon), u_s) \tag{8}$$

which describes the slow dynamics of the system.

Singular perturbations cause a multi-time-scale behaviour of the dynamics characterized by the presence of both slow and fast transients in the system response to external stimuli. Roughly speaking, the slow response or the quasi-steady state is approximated by the reduced model (8) with $\varepsilon = 0$, while the discrepancy between the response of the reduced model (8) with $\varepsilon = 0$ and that of the full model is the fast transient.
To have a closer look at this, let us examine the variable z which has been excluded from the reduced model. By contrast with the original variable z, starting at t_0 from a prescribed z_0, the quasi-steady-state Φ_0 is not free to start from z_0, and there may be a large

discrepancy between its initial value $\Phi_0(x_0, u_s(0))$ and z_0. Thus, Φ_0 is not a uniform approximation of z. However, one expects that the approximation

$$z(t) = \Phi_0(t) + O(\varepsilon) \tag{9}$$

is valid on a time interval excluding t_0, that is for $t \in [t_1,\ T]$ where $t_1 > t_0$.
One can also constraint the quasi-steady state x_s to start from the prescribed initial condition x_0 so that the approximation of x by x_s may be uniform; i.e. one assumes that,

$$x(t) = x_s(t) + O(\varepsilon) \tag{10}$$

holds during a time interval including t_0, that is for $t \in [t_0,\ T]$ over which $x_s(t)$ exists.
The approximation (9) establishes the fact that during an initial boundary layer interval $[t_0,\ t_1]$ the original variable z approaches Φ_0 and then during $[t_1,\ T]$, remains close to Φ_0.

To describe the behaviour of z in a fast-time scale, it is usual to set

$$\tau = \frac{t - t_0}{\varepsilon} \qquad \tau = 0 \quad \text{at} \quad t = t_0; \qquad \eta = z - \Phi_0$$

so that

$$\frac{d\eta}{d\tau} = g(x_0, \eta(\tau) + \Phi_0(t_0), u_s(0) + u_f(\tau)) + O(\varepsilon) \tag{11}$$

with $\eta(t_0) = z_0 - \Phi_0(t_0)$ and x_0, t_0 fixed parameters.

The solution $\eta(\tau)$ to this initial condition value problem is used as a boundary layer correction of (9) for a possibly uniform approximation of z

$$z(t) = \eta(\frac{t - t_0}{\varepsilon}) + \Phi_0(t) + O(\varepsilon) \tag{12}$$

Thus, $\Phi_0(t)$ represents the slow transient of z while $\eta(\tau)$ is the fast transient of z. For ensuring the convergence of (12) to the slow approximation (9), the corrective term $\eta(\tau)$ must decay as $\tau \to \infty$ to some $O(\varepsilon)$ quantity. This is assumed hereafter.

Assumption A. 3 *The equilibrium $\eta = 0$ of (11) is uniformly asymptotically stable in x_0 and t_0 and $\eta(t_0) = z_0 - \Phi_0(t_0)$ belongs to its domain of attraction, moreover $\eta(\tau)$ exists for $\tau \geq 0$ and $\lim_{\tau\to\infty} \eta(\tau) = 0$.*

Assumptions A.3 and A.2 ensure the stability properties of the boundary layer system (11).

It is now possible to state the Tikhonov's Theorem.

Theorem II.1 *([16, 33]) Under assumptions A.2 and A.3, the approximations (10) and (12) are valid for all $t \in [t_0,\ T]$ and there exists $t_1 \geq t_0$ such that (9) is valid for all $t \in [t_1,\ T]$.*

Assuming that $x_s(t)$ is the solution of (8) for $\varepsilon = 0$, we obtain:

$$\begin{aligned}
\frac{d\eta}{d\tau} &= g(x(t), \Phi_0(\tau) + \eta(\tau), u_s + u_f, \varepsilon) - \varepsilon\frac{\Phi_0}{t} \\
&= g(x(t), \Phi_0(\tau) + \eta, u_s + u_f, \varepsilon) - \varepsilon(\frac{\partial\Phi}{\partial x} + \frac{\partial\Phi}{\partial u_s}\,\frac{\partial u_s}{\partial x})f(x, \Phi, u_s) \\
&- \varepsilon(\frac{\partial\Phi}{\partial u_s}\,\frac{\partial u_s}{\partial v})\frac{dv}{dt} \qquad (13)
\end{aligned}$$

which is a more general form of the boundary layer dynamics (11).

II.A A specific class of NLSP Systems

In this subsection, we consider a particular class of NLSP systems which preserves the input linearity after quasi steady state reduction, (see [5, 15, 16, 26, 31, 32]).

This class of systems, denoted Σ_p, is described by the equations

$$\begin{aligned}
\dot{x} &= a_1(x) + a_2(x)z + b_1(x)u \qquad (14)\\
\varepsilon\dot{z} &= a_3(x) + a_4(x)z + b_2(x)u \qquad (15)\\
y &= c_1(x) + c_2(x)z + d(x)u \qquad (16)
\end{aligned}$$

where the functions $a_i(.)$, $b_i(.)$, $c_i(.)$, $d(.)$ are assumed to be real analytic of appropriate dimensions. In what follows, we suppose that Σ_p verifies assumptions A.1 and A.2.

In this specific case, the manifold condition (5) takes the form:

$$\varepsilon \left\{ (\frac{\partial \Phi}{\partial x} + \frac{\partial \Phi}{\partial u}\frac{\partial u}{\partial x}) \left[a_1(x) + a_2(x)\Phi + b_1(x)u \right] + (\frac{\partial \Phi}{\partial u} + \frac{\partial u}{\partial v}\dot{v}) \right\}$$
$$= a_3(x) + a_4(x)\Phi + b_2(x)u \qquad (17)$$

Substituting the expansion (7) into (17) and equating terms of the same powers in ε (up to ε^2), gives for example the solutions

$$\Phi_0 = -{a_4}^{-1}(x)\left\{a_3(x) + b_2(x)u\right\} \qquad (18)$$

and

$$\Phi_1 = {a_4}^{-1}(x)\left\{ \tfrac{\partial \Phi_0}{\partial x}(a_1(x) + a_2(x)\Phi_0 + b_1(x)u) + \tfrac{\partial \Phi_0}{\partial u}\dot{u}_s \right\} \qquad (19)$$

where $\dot{u_s} = \frac{\partial u}{\partial x}\dot{x} + \frac{\partial u}{\partial v}\dot{v}$.

According to (8), the reduced slow system Σ_p^1, approximated at the order 1 in ε, is described by:

$$\begin{aligned} \dot{x} &= a_1(x) + a_2(x)\,(\Phi_0 + \varepsilon\Phi_1) + b_1(x)u \\ &= \left\{a_1(x) - a_2(x){a_4}^{-1}(x)a_3(x)\right\} \\ &+ \left\{b_1(x) - a_2(x){a_4}^{-1}(x)b_2(x)\right\} u + \varepsilon a_2(x)\Phi_1 \end{aligned} \qquad (20)$$

and the output is given by

$$y = c_1(x) + c_2(x)(\Phi_0 + \varepsilon\Phi_1)u + d(x)u + O(\varepsilon^2) \qquad (21)$$

where $(x_s,\ u_s)$ is replaced by $(x,\ u)$.
Setting,

$$\begin{aligned} f^0(x) &= a_1(x) - a_2(x){a_4}^{-1}(x)a_3(x) \\ g^0(x) &= b_1(x) - a_2(x){a_4}^{-1}(x)b_2(x) \end{aligned} \qquad (22)$$

$$\begin{aligned} h^0(x) &= c_1(x) - c_2(x)a_4{}^{-1}(x)a_3(x) \\ l^0(x) &= d(x) - c_2(x)a_4{}^{-1}(x)b_2(x) \end{aligned}$$

and

$$\begin{aligned} p^1(x, u, \dot{u}, ...) &= a_2(x)\Phi_1 \\ h^1(x, u, \dot{u}, ...) &= c_2(x)\Phi_1 \end{aligned}$$

we get, at the order 1 in ε, the compact expression of Σ_p^1:

$$\begin{aligned} \dot{x} &= f^0(x) + g^0(x)u + \varepsilon p^1(x, u, \dot{u}, ...) \\ y &= h^0(x) + l^0(x)u + \varepsilon h^1(x, u, \dot{u}, ...) \end{aligned}$$

where $\dot{u}, \ddot{u}, ...$ denote the successive derivatives of u.

Neglecting the terms in $O(\varepsilon)$, we obtain the *steady-state reduced dynamics*

$$\Sigma_p^0 = \begin{cases} \dot{x} = f^0(x) + g^0(x)u \\ y = h^0(x) + l^0(x)u \end{cases} \tag{23}$$

which is linear with respect to u. f^0, g^0, h^0 and l^0 are given by the unperturbed system (22).

Remark: When $d(x) = 0$, Σ_p is said strictly causal. This property is lost for the steady state reduced system Σ_p^0, if $c_2(x)a_4^{-1}(x)b_2(x) \neq 0$.

III Sampled Schemes for NLSP Dynamics

In this section we recall a sampled scheme proposed in [4], for the nonlinear singularly perturbed dynamics described by equations (1) and (2). i.e.,

$$D_{sp} = \begin{cases} \dot{x} = f(x, z, u) \\ \varepsilon\dot{z} = g(x, z, u) \end{cases} \tag{24}$$

that we rewrite as

$$\dot{Y} = (F + \frac{G}{\varepsilon}) \tag{25}$$

where $Y = \begin{pmatrix} x \\ z \end{pmatrix}$, $F = \begin{pmatrix} f \\ 0 \end{pmatrix}$ and $G = \begin{pmatrix} 0 \\ g \end{pmatrix}$

Let δ be a fixed sampling period. For a piecewise constant control over time intervals of amplitude δ,

$$u(t) := u(p), \quad p\delta \leq t < (p+1)\delta, \ p \geq 0 \tag{26}$$

starting from $Y(p)$, the solution to (25) at time $t = (p+1)$ is given by the Lie series

$$Y(p+1) = e^{\delta L_{(F+\frac{G}{\varepsilon})}} Id_{|Y(p)} \tag{27}$$

where the vectors fields F and G depend on $u(p)$. (27) is detailed in Appendix 2.

Hereafter, we will distinguish between fast sampling, $\delta = \alpha\varepsilon$, with α close to 1 and slow sampling $\delta = \beta\varepsilon$ with δ fixed and ε going to 0^+ as β goes to $+\infty$. The integers n and k will respectively, denote the fast and slow sampling times.

III.A Fast Sampling

In this subsection, the fast sampling of the nonlinear dynamics D_{sp}, is considered. Assuming the sampling period δ sufficiently close to ε, one writes δ as $\alpha\varepsilon$ where α is a real number close to 1. In the linear context [11, 17], α is usually equal to 1, this particular choice is not made here. The fast sampled model is used to assign the fast closed-loop behaviour or to stabilize the fast state component z when assumption A.2 is not satisfied.

For $\delta = \alpha\varepsilon$ and reminding that $e^{-\alpha L_G} e^{\alpha L_G} = Id$, (27) can be decomposed as

$$Y(n+1) = \{e^{\alpha L_{\varepsilon F+G}} e^{-\alpha L_G}\} e^{\alpha L_G} I_{d|Y(n)}$$

which can be expanded in powers of ε involving the Baker-Campbell-Hausdorff formula as in [19].

Theorem III.1 *[4]: The exact fast discrete-time dynamics associated to D_{sp}, is given by:*

$$Y(n+1) = \{I_d + \sum_{k\geq 1} \frac{\varepsilon^k}{k!} \sum_{(i_m+\ldots+i_1)=k} \tilde{C}(i_m, \ldots, i_1)$$

$$\tilde{E}^{i_m}(G, F^{i_m}) \ldots \tilde{E}^{i_1}(G, F^{i_1})\} e^{\alpha L_G} I_{d|Y(n)} \tag{28}$$

where the $\tilde{E}^i(G, F^i)(I_d)$ are vector fields belonging to the Lie algebra generated by F and G, iteratively defined as:

$$\tilde{E}^1(G, F) := \frac{d(e^{L_{G+tF}} e^{-L_G})}{dt}_{|t=0} \tag{29}$$

$$\tilde{E}^{i+1}(G, F^{i+1}) := \frac{d(\tilde{E}^i(G + tF, F^i))}{dt}_{|t=0}, \; i \geq 1 \tag{30}$$

The coefficients $\tilde{C}(i_m, ..., i_2, 1)$ are integers iteratively defined by

$$\begin{aligned} \tilde{C}(i_m, ..., i_2, 1) &= \tilde{C}(i_m, ..., i_2) \\ &+ \tilde{C}(i_m, ..., i_2 - 1, 1) + ... + \tilde{C}(i_m - 1, ..., i_2, 1) \\ \textit{with} \quad & \quad \tilde{C}(i_m, ..., i_2, i_1) = 0 \qquad \forall i_j \leq 0 \\ \textit{and} \quad & \quad \tilde{C}(i) = 1 \qquad \forall i \geq 1 \end{aligned}$$

Proof: Theorem III.1 is a consequence of a result obtained in [19] concerning the general representation of the sampled dynamics in terms of the vectors fields $\tilde{E}^i$, the proof is given in Appendix 3. □

Even if computational difficulties occur, the advantage of (28) is to give a precise expression of the fast dynamics at any order of approximation with respect to ε. In many cases, approximation at the order 1 is sufficient so that a simplified model is computed as specified in the next corollary.

Corollary III.1 *The first order approximation in ε of the fast discrete-time dynamics (28) is given by*

$$Y(n+1) = e^{\alpha L_G} I_{d|Y(n)} + \varepsilon \tilde{E}^1(\alpha G, \alpha F) e^{\alpha L_G} I_{d|Y(n)}$$

which can be decomposed as:

$$\begin{aligned} x(n+1) &= x(n) + \varepsilon \tilde{E}^1(\alpha G, \alpha F) I_{d_{ns}|Y(n)} \\ z(n+1) &= \left\{ e^{\alpha L_G} I_{d_{nf}} + \varepsilon \tilde{E}^1(\alpha G, \alpha F) e^{\alpha L_G} I_{d_{nf}} \right\}_{|Y(n)} \end{aligned} \tag{31}$$

$I_{d_{ns}}$ and $I_{d_{nf}}$ denote the identity projections from R^n to R^{ns} or R^{nf} respectively.
More precisely, we have:

$$\begin{aligned}
\tilde{E}^1(\alpha G, \alpha F) I_{d_{ns}|Y(n)} &= \sum_{i\geq 1} \frac{\alpha^i}{i!} L_G^{i-1}(f(x,z,u)). \\
e^{\alpha L_G} I_{d_{nf}|Y(n)} &= e^{\alpha L_{g(x,z,u)}}(z)_{|Y(n)}
\end{aligned}$$

Equation (31) results from the fact that, because of the structure of G in (25), one has

$$e^{\alpha L_G}(I_{d_{ns}}) \mid_{Y(n)} = x(n)$$

The above expressions clearly put in light the distinction, under fast sampling, between the slow component, x, which evolves in $O(\varepsilon)$ and the fast component z.

The dynamics

$$\begin{aligned}
x(n+1) &= x(n) + \varepsilon \tilde{E}^1(\alpha G, \alpha F) I_{d_{ns}|Y(n)} \\
z(n+1) &= e^{\alpha L_G}(I_{d_{nf}})
\end{aligned}$$

is called the "dominant fast sampled dynamics" in the sense that it emphasizes the fast component dynamics in the present context of fast sampling.

The Euler approximation of (24) is

$$\begin{aligned}
x(k+1) &= x(k) + \varepsilon \alpha f(x(k), z(k), u(k)) \\
z(k+1) &= z(k) + \alpha g(x(k), z(k), u(k))
\end{aligned}$$

which is simply the first order approximation with respect to α of the dominant fast sampled dynamics. Such an approximation may be strongly dammageable for the fast dynamics evolution. In fact, the Euler approximation does not take into account the two-time-scale property of the initial system.

Remark: In the text a_i^k [resp. a_i^n] denotes $a_i(x(k))$ [resp $a_i(x(n))$] and b_i^k [resp. b_i^n] denotes $b_i(x(k))$ [resp $b_i(x(n))$].

Hearafter, some examples are performed.

III.A.1 A specific class of NLSP dynamics

Considering the dynamics associated to the system Σ_p, (14)-(15), we have:

$$F = \begin{pmatrix} a_1(x) + a_2(x)z + b_1(x)u \\ 0 \end{pmatrix}$$
$$\text{and} \qquad G = \begin{pmatrix} 0 \\ a_3(x) + a_4(x)z + b_2(x)u \end{pmatrix}$$

Thus, from Corollary III.1:

$$e^{\alpha L_G} I_{d_{nf}|Y(n)} = z(n) + \alpha[a_3^n + a_4^n z + b_2^n u]$$
$$-\frac{\alpha^2}{2!} a_4^n [a_3^n + a_4^n z + b_2^n u] + O(\alpha^3)$$

and

$$\tilde{E}^1(\alpha G, \alpha F) I_{d_{ns}|Y(n)} = \alpha[a_1^n + a_2^n z + b_1^n u(n)] +$$
$$\frac{\alpha^2}{2!} a_2^n (a_3^n + a_4^n z + b_2^n u(n)) + O(\alpha^3)$$

Finally, from Corollary III.1, we obtain the dominant fast sampled dynamics

$$\begin{aligned} x(n+1) &= x(n) + \varepsilon\{\alpha(a_1^n + a_2^n z + b_1^n u(n)) \\ &\quad + \frac{\alpha^2}{2!} a_2^n (a_3^n + a_4^n z + b_2^n u(n)) + O(\alpha^3)\} \\ z(n+1) &= z(n) + \alpha(a_3^n + a_4^n z + b_2^n u(n)) \\ &\quad + \frac{\alpha^2}{2!} a_4^n (a_3^n + a_4^n z + b_2^n u(n)) + O(\alpha^3) \end{aligned}$$

III.A.2 Example: A nonlinear dynamics

Consider the simple nonlinear dynamics:

$$\Sigma_2 = \begin{cases} \dot{x} = z \\ \varepsilon \dot{z} = -x - \tan z + u \end{cases} \tag{32}$$

From Corollary III.1, we get:

$$e^{\alpha L_G} I_{d_{nf}|Y(n)} = z(n) + \alpha(-x(n) - \tan z(n) + u(n))$$
$$-\frac{\alpha^2}{2!}(1 + (\tan z(n))^2)(-x(n) - \tan z(n) + u(n)) + O(\alpha^3)$$

and:

$$\tilde{E}^1(\alpha G, (\alpha F)) I_{d_{ns}|Y(n)} = \alpha z(n) +$$
$$\frac{\alpha^2}{2!}(-x(n) - \tan z(n) + u(n)) + O(\alpha^3)$$

Rewriting these approximations in a compact form yields to the dominant fast sampled dynamics

$$\begin{aligned} x(n+1) &= x(n) + \varepsilon[\alpha z(n) \\ &\quad + \frac{\alpha^2}{2!}(-x(n) - \tan z(n) + u(n)) + O(\alpha^3)] \\ z(n+1) &= z(n) + \left\{\alpha - \frac{\alpha^2}{2!}[1 + (\tan z(n))^2)]\right\} \\ &\quad (-x(n) - \tan z(n) + u(n) + O(\alpha^3)) \end{aligned}$$

III.A.3 The linear case of NLSP dynamics

Consider the linear singularly perturbed dynamic D_l defined by

$$D_l := \begin{cases} \dot{x} = A_{11}x + A_{12}z + B_1 u \\ \varepsilon\dot{z} = A_{21}x + A_{22}z + B_2 u \end{cases}$$

where the matrices A_i and B_i are of appropriate dimensions. In this case, we have

$$Y = \begin{pmatrix} x \\ z \end{pmatrix}, \quad F = \begin{pmatrix} A_{11}x + A_{12}z + B_1 u \\ 0 \end{pmatrix}$$
$$\text{and} \quad G = \begin{pmatrix} 0 \\ A_{21}x + A_{22}z + B_2 u \end{pmatrix}$$

After some linear algebraic manipulations, we verify that

$$e^{\alpha L_G} I_{d_{nf}|Y(n)} = z(n) + \sum_{i=1}^{\infty} \frac{\alpha^i}{i!}[A_{22}^i z(n) + A_{22}^{i-1}(A_{21}x(n) + B_2 u(n))] \quad (33)$$

which can be exactly rewritten as

$$e^{\alpha L_G} I_{d_{nf}|Y(n)} = e^{\alpha A_{22}} z(n) + \frac{e^{\alpha A_{22}} - I_d}{A_{22}}(A_{21}x(n) + B_2 u(n)) \quad (34)$$

and for $\tilde{E}^1(\alpha G, \alpha F) I_{d_{ns}|Y(n)}$, we get:

$$\begin{aligned} \tilde{E}^1(\alpha G, \alpha F) I_{d_{ns}|Y(n)} &= \alpha(A_{11}x(n) + B_1 u(n)) + \alpha A_{12} & (35)\\ & \quad \{(z(n) + \sum_{i=2}^{\infty} \frac{\alpha^{i-1}}{i!} A_{22}^{i-2}(A_{21}x(n) + A_{22}z(n) + B_2 u(n))\} \\ &= \alpha(A_{11}x(n) + B_1 u(n)) & (36)\\ &+ A_{12}\frac{e^{\alpha A_{22}} - I_d - \alpha A_{22}}{A_{22}^2}(A_{21}x(n) + A_{22}z(n) + B_2 u(n)) \end{aligned}$$

The dominant sampled dynamics associated to D_l takes the form

$$\begin{pmatrix} x(n+1) \\ z(n+1) \end{pmatrix} = \begin{pmatrix} I_d + \varepsilon \tilde{A}_{11} & \varepsilon \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} \begin{pmatrix} x(n) \\ z(n) \end{pmatrix} + \begin{pmatrix} \varepsilon \tilde{B}_1 \\ \tilde{B}_2 \end{pmatrix} u(n) \quad (37)$$

where

$$\begin{aligned} \tilde{A}_{11} &= \alpha(A_{11} - A_{12}A_{22}^{-1}A_{21}) + A_{12}A_{22}^{-1}(e^{\alpha A_{22}} - I)A_{22}^{-1}A_{21} \\ \tilde{A}_{12} &= A_{12}A_{22}^{-1}(e^{\alpha A_{22}} - I) \\ \tilde{A}_{21} &= (e^{\alpha A_{22}} - I)A_{22}^{-1}A_{21} \\ \tilde{A}_{22} &= e^{\alpha A_{22}} \\ \tilde{B}_1 &= \alpha(B_1 - A_{12}A_{22}^{-1}B_2) + A_{12}A_{22}^{-1}(e^{\alpha A_{22}} - I)A_{22}^{-1}B_2 \\ \tilde{B}_2 &= (e^{\alpha A_{22}} - I)A_{22}^{-1}B_2 \end{aligned}$$

Remarks :

i) The invertibility of the matrix A_{22} is not required but only used to write combinatoric expansions in (34) and (36). If A_{22} is not invertible one uses the expressions (33) and (35).

ii) The solution (37) is the fast dynamics proposed in [11, 17] making use of a state space decoupling transformation not available in a nonlinear context.
iii) For the linear case, the Euler approximation gives

$$x(n+1) \simeq x(n) + \delta(A_{11}x(n) + A_{12}z(n) + B_1u(n))$$

$$z(n+1) \simeq z(n) + \delta\frac{(A_{21}x(n) + A_{22}z(n) + B_2u(n))}{\varepsilon}$$

which clearly differs from (37). The Euler approximation exhibits less informations about the inherent structure of the system.

III.B Slow Sampling

Since the slow sampling period δ means that $\delta \gg \varepsilon$, let us write δ as $\beta\varepsilon$. This transformation simplifies the reduction operation after slow sampling. Since $\delta = \beta\varepsilon$, the formula (27) becomes

$$Y(k+1) = e^{L_{(\delta F + \beta G)}} I_{d|Y(k)} \tag{38}$$

with F and G depending on $u(k)$ and which can be decomposed again as

$$Y(k+1) = e^{\beta L_G}\{e^{-\beta L_G} e^{\beta L_{G+\varepsilon F}}\} I_{d|Y(k)}$$

The next theorem gives an explicit expression of the solution (38) in terms of integro differential Poincaré forms proposed in [20]. This allows the study of the behaviour of the solution as ε goes to zero or equivalently β goes to infinity (the reduction operation).

Theorem III.2 *[4]: The slow sampled dynamics associated to (25), is given by*

$$Y(k+1) = e^{\beta L_G} I_{d|Y(k)} + \delta \int_0^1 e^{(1-t)\beta L_G} L_F e^{t\beta L_G} I_{d|Y(k)} dt$$

$$+ \delta^2 \int_0^1 \int_0^t (1-t) e^{(1-t)\beta L_G} L_F e^{(t-\sigma)\beta L_G} L_F e^{\sigma\beta L_G} I_{d|Y(k)} d\sigma dt + O(\delta^3) \tag{39}$$

Proof: The proof is sketched in Appendix 3, by substituting t for $\beta\varepsilon$, P for βG and Q for F. □

The slow sampled dynamics is used to design the slow controller assuming the stability of the fast dynamics. For this reason it is natural to look for a reduced slow sampled dynamics on the basis of which the slow control computation is simplified. This is developed in the next subsection.

III.C Reduction of the slow sampled dynamics

In this subsection, we assume that the dynamics D_{sp} verifies the two assumptions A.2 and A.3.
Because δ is fixed, to take the limit as ε goes to 0^+ is equivalent to take the limit as β goes to $+\infty$. Consequently, the reduced slow sampled dynamics is deduced from (38) just taking the limit as β goes to $+\infty$. This underlines the interest of the representation (38).

Proposition III.1 *Given the continuous-time dynamics (25), under assumptions A.2 and A.3, as $\varepsilon \to 0^+$, one gets the following reduced slow sampled dynamics*

$$\begin{aligned} Y(k+1) &= e^{\delta L_{F(Y)}}(\bar{Y})_{|\bar{Y}(k)} \\ &= \bar{Y}(k) + \delta(L_F\bar{Y})_{|\bar{Y}(k)} \\ &\quad + \frac{\delta^2}{2!} L_F(L_F\bar{Y}_{|\bar{Y}(k)})_{|\bar{Y}(k)} + O(\delta^3) \end{aligned} \tag{40}$$

where $\bar{Y} = (x^T, \bar{z}^T)^T$ and $\bar{z} = \Phi_0(x,u)$ is the solution of (6).

Sketch of Proof The proof, developed in [4], follows from the fact that due to the form of G, in (25), the limit as β goes to $+\infty$ of $e^{\beta L_G}(I_d)_{|Y(k)}$ is the solution at time $\beta = \frac{\delta}{\varepsilon}$ of the dynamics

$$\begin{aligned} \dot{x} &= 0 \\ \dot{z} &= g(x, z, u(k)) \end{aligned}$$

with a constant input, $u(k)$, over δ.

Because of the stability assumptions A.2 and A.3, we get

$$\lim_{\beta \to +\infty} e^{\beta L_G}(I_d)_{|Y(k)} = (x^T(k), \bar{z}^T(k))^T \stackrel{def}{=} \bar{Y}(k)$$

whith $\bar{z}(k) = \Phi_0(x(k), u(k))$ the solution of (6).

An easy iterative reasonning enables to conclude (40) where $L_F(\bar{Y})_{|\bar{Y}(k)}$ stands for

$$L_F(x^T, \bar{z}^T(x,u))_{|\bar{Y}(k)} = \left\{ f^T(x,\bar{z}), \left(\frac{\partial \bar{z}}{\partial x} f(x,\bar{z}) \right)^T \right\}_{|\bar{Y}(k)}$$

and so on for the successive compositions of Lie derivatives appearing in (40). □

In order to facilitate the control design, the next corollary precises the decomposition of (40) into slow and fast components.

Corollary III.2 *The decomposition into the slow and fast components of the reduced slow sampled dynamics (40), is*

$$\begin{aligned} x(k+1) &= x(k) + \delta f(x,\bar{z})_{|\bar{Y}(k)} \\ &\quad + \frac{\delta^2}{2} \left\{ \frac{\partial f(.,\bar{z})}{\partial x}) + \left(\frac{\partial f(x,.)}{\partial \bar{z}} \frac{\partial \bar{z}}{\partial x} \right) f(.) \right\}_{|\bar{Y}(k)} + O(\delta^3) \end{aligned} \tag{41}$$

$$z(k+1) = \bar{z}(k) + \delta \frac{\partial \bar{z}}{\partial x} f(.)_{|\bar{Y}(k)} + \frac{\delta^2}{2} L_f \left(\frac{\partial \bar{z}}{\partial x} f(.) \right)_{|\bar{Y}(k)} + O(\delta^3)$$

Remarks

- The slow-time scale model exhibits the slow dynamics in x used to compute a slow controller as well as the evolution of the quasi-steady state.
- In equation (41), $z(k+1)$ does not depend on z and is only function of x and u. Thus, the reduced slow sampled dynamics is of dimension n_f.
- (41) may be interpreted as the solution of

$$\dot{x} = f(x, \bar{z}(x,u), u)$$

for the slow dynamics. Consequently the reduction before or after sampling gives the same result for the slow state components.

Since $\bar{z}$ in (41) is function of u, a critical question is if it depends on $u(k)$ or $u(k-1)$, this will be discussed in Section IV in the context of controlling the input output behaviour.

III.C.1 A specific class of NLSP dynamics

Consider the dynamics associated to the system Σ_p, (14)-(15), with $a_4(x)$ stable and regular. Applying Corollary III.2, we obtain the reduced slow sampled dynamics

$$\begin{aligned} x(k+1) &= x(k) + \delta(a_1^k + a_2^k \bar{z}(k) + b_1^k u(k)) \\ &+ \frac{\delta^2}{2!}\left(\frac{\partial a_1(x)}{\partial x} + \frac{\partial a_2(x)}{\partial x}\bar{z}(k) + \frac{\partial b_1(x)}{\partial x}u(k) + a_2^k\frac{\partial \bar{z}}{\partial x}\right)_{|x(k)} \\ &\left(a_1^k + a_2^k \bar{z}(k) + b_1^k u(k)\right) + O(\delta^3) \end{aligned} \tag{42}$$

$$\begin{aligned} z(k+1) &= \bar{z}(k) + \delta\frac{\partial \bar{z}}{\partial x}(a_1^k + a_2^k \bar{z}(k) + b_1^k u(k)) \\ &+ \frac{\delta^2}{2!}\frac{\partial}{\partial x}\left(\frac{\partial \bar{z}}{\partial x}(a_1^k + a_2^k \bar{z}(k) + b_1^k u(k))\right)_{|x(k)} \\ &\left(a_1^k + a_2^k \bar{z}(k) + b_1^k u(k)\right) + O(\delta^3) \end{aligned} \tag{43}$$

where $\bar{z}(x,u) = -[a_4(x)]^{-1}(a_3(x) + b_2(x)u)$

In (42), it is important to note that the term $a_2(x)\frac{\partial \bar{z}}{\partial x}$ characterizes the influences over $x(k+1)$ of the evolution of the state $\bar{z}$ on the slow manifold.

III.C.2 Example: A nonlinear dynamics

For the nonlinear dynamics defined in example III.A.2, the quasi steady state solution $\bar{z}$ is given by

$$\bar{z} = -tan^{-1}(u-x)$$

and the reduced slow sampled model is

$$x(k+1) = x(k) + \delta\bar{z}(k) + O(\delta^2)$$

$$z(k+1) = \bar{z}(k)[1 + \frac{\delta}{1+(u(k)-x(k))^2}] + O(\delta^2)$$

III.C.3 The linear case of NLSP dynamics

Consider again the singularly perturbed linear system of example III.A.3 with A_{22} stable and invertible. Applying Theorem III.2 we obtain

$$\begin{aligned} x(k+1) &= x(k) + \delta\{A_{11}x(k) + B_1u(k) + A_{12}(e^{\beta A_{22}}z(k) \\ &\quad + \frac{e^{\beta A_{22}}-1}{A_{22}}(A_{21}x(k) + B_2u(k)))\} + O(\delta^2) \\ z(k+1) &= e^{\beta A_{22}}z(k) + \frac{e^{A_{22}\beta}-1}{A_{22}}(A_{21}x(k) + B_2u(k))) + O(\delta) \end{aligned}$$

As $\beta = \frac{\delta}{\epsilon}$, we set $\lim_{\beta\to+\infty} e^{\beta A_{22}} = 0$ and applying Corollary III.2, with $\bar{z}$ given by

$$\bar{z} = -A_{22}^{-1}(A_{21}x + B_2u)$$

we obtain

$$\begin{aligned} x(k+1) = x(k) + \delta(A_{11}x(k) + A_{12}\bar{z}(k) + B_1u(k)) \\ +\frac{\delta^2}{2!}(A_{11} - A_{12}A_{22}^{-1}A_{21})(A_{11}x(k) + A_{12}\bar{z}(k) + B_1u(k)) + O(\delta^3) \end{aligned} \quad (44)$$

where the fast state component z is described by:

$$\begin{aligned} z(k+1) &= -A_{22}^{-1}(A_{21}x(k) + B_2u(k)) - \\ &\quad \delta A_{22}^{-1}A_{21}(A_{11}x(k) + A_{12}\bar{z}(k) + B_1u(k)) - \\ &\quad \frac{\delta^2}{2!}A_{22}^{-1}A_{21}(A_{11} - A_{12}A_{22}^{-1}A_{21}) \\ &\quad (A_{11}x(k) + A_{12}\bar{z}(k) + B_1u(k)) + O(\delta^3) \end{aligned} \quad (45)$$

The next section studies in details, the non commutativity between reduction and sampling for input output behaviour.

IV Reduction and Sampling of NLSP Systems

It is well known that, in the linear case ([3, 7]), Reduction before Sampling (RS) and Slow Sampling before Reduction (SR) do not give the same results when input-output behaviours are studied.
In particular, under RS, undesirable zeroes appear under sampling

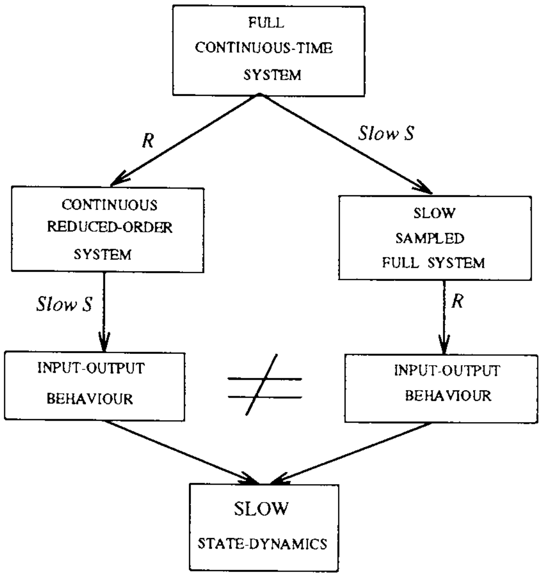


Figure 1 : Commutativity of reduction (R) and Sampling (S) procedures

in the transfer function, because a direct link may be present in the continuous-time transfer function associated to the reduced system. However, in practice the procedure RS is used since the reduced continuous-time dynamics only is available. We discuss this problem in a nonlinear context.

For technical reasons: linearity in u and slow manifold computation, we limit our study to the class of NLSP systems of the form Σ_p (equations (14)-(16)), where the matrix $a_4(x)$ is assumed to be regular and stable.

Due to the sampling procedure and the fact that the quasi-steady state solution $\bar{z}$ depends on u, it is necessary, at each time instant $k\delta$, to distinguish between $\bar{z}^-(k)$ and $\bar{z}^+(k)$ which represent $\bar{z}(k)$ just before and just after the jump respectively (see figure 2).

From the result of subsection III.B we immediately deduce

Lemma IV.1 *The solution of* $\Phi_0(x,u)$ *just before the sampling time* $k\delta$, $\bar{z}^-(k)$, *is:*

$$\bar{z}^-(k) = -[a_4^k]^{-1}[a_3^k + b_2^k \boldsymbol{u}(k-1)] \tag{46}$$

Proof: The proof is achieved taking the limit as t goes to $k\delta + \delta^-$ of x in equation (42)

$$\begin{aligned} x^-(k+1) &= x(k) + \delta\{a_1^k - a_2^k[a_4^k]^{-1}(a_3^k + b_2^k \boldsymbol{u}(k)) \\ &\quad + b_1^k \boldsymbol{u}(k)\} + O(\delta^2) \end{aligned} \tag{47}$$

Moreover, the limit of $u(t)$ as t goes to $k\delta + \delta^-$ is equal to $u(k)$. Thus from (18) we find

$$\bar{z}^-(k+1) = -[a_4^{k+1}]^{-1}(a_3^{k+1} + b_2^{k+1}\boldsymbol{u}(k)) \tag{48}$$

□

Similary, we have:

Lemma IV.2 *The solution of* $\Phi_0(x,u)$ *just after the sampling time* $k\delta$, $\bar{z}^+(k)$, *is:*

$$\bar{z}^+(k) = -[a_4^k]^{-1}[a_3^k + b_2^k \boldsymbol{u}(k)] \tag{49}$$

Proof: For $z(k)$ at time $k\delta$, and under the assumptions A.1 and A.2, we obtain for any fixed short period of time close to 0^+, $\delta_\tau = \rho\varepsilon > 0$

$$\begin{aligned} z(k+\rho) &= \lim_{\rho\to+\infty} e^{\rho L_F} I_{d_{nf}}{}_{|Y(k)} \\ &= \lim_{\rho\to\infty} e^{\varepsilon\rho L_{(a_3^k + a_4^k z(k) + b_2^k \boldsymbol{u}(\mathbf{k}))}} z(k) \\ &= -[a_4^k]^{-1}[a_3^k + b_2^k \boldsymbol{u}(k)] \end{aligned}$$

thus

$$\bar{z}^{+}(k) = -[a_4^k]^{-1}[a_3^k + b_2^k \boldsymbol{u}(k)]$$

□

Comparing equation (46) and (49), we immediately see that $\bar{z}^{-}(k)$ is

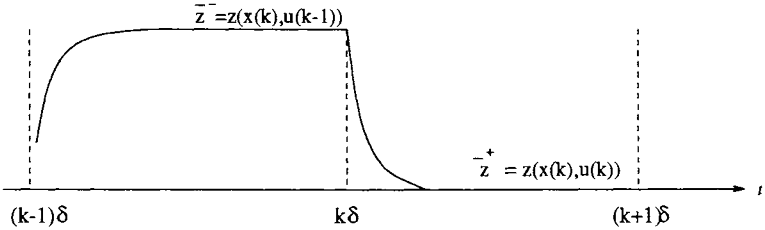


Figure 2 : $\bar{z}$

different from $\bar{z}^{+}(k)$ if $\boldsymbol{u}(k-1)$ is different from $\boldsymbol{u}(k)$. This is the key point of the proposed control in Section V and the results of the two next subsections is a direct consequence of this.

IV.A Reduction of the slow sampled system

Let us now compute the system obtained by reduction after slow sampling of Σ_p. It is important to note that it is the correct scheme. The input sampling and the data acquisition being strictly synchronized, the fast dynamics have no time to converge to the slow manifold parametrized by $\boldsymbol{u}(k+1)$ as it is assumed when reduction is performed before sampling. As pointed in the proof of the next proposition, this is due to the difference between $\bar{z}^{-}(k)$ and $\bar{z}^{+}(k)$. It is important to note also that at time $k\delta$ the output is expressed in term of $\bar{z}^{-}(k)$

Proposition IV.1 *The reduced slow discrete-time system, noted Σ_{DR}, is given by*

$$\begin{aligned}
x(k+1) &= e^{\delta L_{\{a_1(.)+a_2(.)\bar{z}(.,u(k))+b_1(.)u(k)\}}} I_{n_s|x(k)} \\
&= x(k) + \delta(a_1^k + b_1^k \boldsymbol{u}(k)) \qquad (50)\\
&\quad + \delta a_2^k\{-[a_4^k]^{-1}(a_3^k + b_2^k \boldsymbol{u}(k))\} + O(\delta^2) \\
y(k) &= c_1^k - c_2^k[a_4^k]^{-1}\{a_3^k + b_2^k \boldsymbol{u}(k-1)\} + d^k \boldsymbol{u}(k) \qquad (51)
\end{aligned}$$

Proof: Equation (50) is immediately given by Corollary III.2. Moreover, equation (40) of Theorem III.2 applied to h instead of I_d, gives:

$$\begin{aligned} y(k+1) &= h(\bar{Y}(k)) + \delta L_F h(\bar{Y})_{|\bar{Y}(k)} \\ &\quad + \frac{\delta^2}{2!} L_F(L_F h(\bar{Y}))_{|\bar{Y}(k)} + O(\delta^3) \end{aligned}$$

with $h(\bar{Y}) = c_1(x) + c_2(x)\bar{z} + d(x)u$.
Using the lemma IV.2 and arguing as in its proof, we obtain:

$$y(k) = c_1^k - c_2^k[a_4^k]^{-1}\{a_3^k + b_2^k \boldsymbol{u}(k-1)\} + d^k \boldsymbol{u}(k)$$

□

As it is pointed out in the previous section, the dynamics of z is not dynamics one after reduction because any state is function of the initial value of z. This is also true for sampling after reduction because the reduced system have not a dynamics in z.

IV.B Sampling of the reduced system

Usually, only the steady state slow continuous-time system (23) is available. This is generally due to the fact that high-frequency dynamics is unknown. Let a system (23) and assume a constant control defined in (26) with a slow sampling period assumed $\delta = \beta\varepsilon \gg \varepsilon$, we obtain

Proposition IV.2 *The slow sampling of the reduced system (23) gives to the following system, noted* Σ_{RD}:

$$\begin{aligned} x(k+1) &= e^{\delta L_{\{a_1(\cdot)+a_2(\cdot)\bar{z}(\cdot,u(k))+b_1(\cdot)u(k)\}}} I_{n_s|x(k)} \\ &= x(k) + \delta(a_1^k + b_1^k \boldsymbol{u}(k)) \qquad (52) \\ &\quad + \delta a_2^k\{-[a_4^k]^{-1}(a_3^k + b_2^k \boldsymbol{u}(k))\} + O(\delta^2) \\ y(k) &= c_1^k - c_2^k[a_4^k]^{-1}\{a_3^k + b_2^k \boldsymbol{u}(\mathbf{k})\} + d^k \boldsymbol{u}(\mathbf{k}) \qquad (53) \end{aligned}$$

The input state behaviours (50) and (52) are the same, i.e, the diagram commutes, because during the sampling period $[k\delta\ (k+1)\delta[$ $\boldsymbol{u}(k)$ only acts on the dynamics and not $\boldsymbol{u}(k-1)$, the boudary layer influence is neglected. But with respect to the output map, in the RS procedure the fast dynamics are first replaced by a quasi-state solution Φ_0. This introduces a fictitious direct link between the input and output. So, during a data acquisition of $y(k)$ one neglects the small delay due to the fast dynamics. In the SR procedure this delay is not neglected, because one reduces the discrete-time system. This comparative study is summarized in figure 1. Finally, $\bar{z}^{-}(k)$ appears explicitly in the solution $y(k)$.

V Control scheme for NLSP systems with-UHFD

In this section, a control scheme for NLSP systems with Unmodelled High-Frequency Dynamics (UHFD) is presented. The goal of the control scheme depicted in figure 3 is to ensure the equality between (53) and (51) at time $(k+1)$. Recalling that the continuous-time reduced dynamics is the only one known by the control designer, then the RS procedure must be used in the control scheme.

Consider the sampled data-control scheme (delayed zero order hold) where the input variable $u(t)$ is given as:

$$u(t) = \begin{cases} u_k^1 & \text{for } kT \ < t \ \le (k+\alpha)T \\ u_k^2 & \text{for } (k+\alpha)T \ < t \ \le (k+1)T \end{cases} \tag{54}$$

with $u_k^1 = u_{k-1}^2$ (see figure 3) and suitable choice for α. More precisely

Assumption A. 4 *The control law (54) depicted in figure 3, is such that δ and α satisfy*

$$(1-\alpha)\delta \gg \varepsilon \ \ \textit{and} \ \ \alpha\delta \gg \varepsilon$$

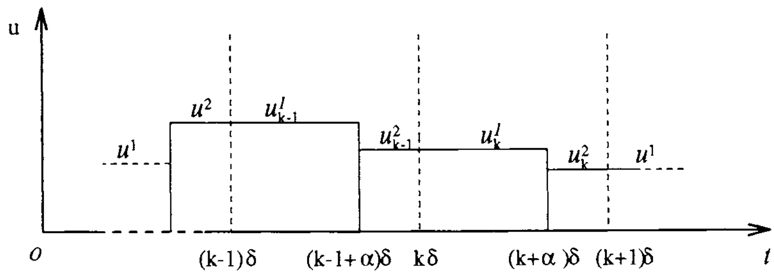


Figure 3 : the control scheme u

This assumption guarantees that $\alpha\delta$ and $(1-\alpha)\delta$ may be considered as slow sampling rates and that at any time $(k+1)\delta$ the fast state η is close to 0 and subsequently the reduction procedure is allowed. Assumption A.4 guarantees also a satisfactory data measurement at the sampling times. Roughly speaking the output data acquisitions are prohibited during the jumps of the integral manifold.

This kind of control scheme is often used in practice because all the operations are not realized exactly at the same time by the computer as also studied in [22] in the context of sampling systems with delay. This study shows that it is important to realize the data acquisition before changing the input.

V.A Reduction of the sampled system with a delayed zero-order hold

Under the input configuration (54), composing the dynamics (50) over two intervals of amplitude $\alpha\delta$ and $(1-\alpha)\delta$ we get

$$\begin{aligned} x(k+1) &= x(k) + \delta a_1^k - \delta a_2^k [a_4^k]^{-1} a_3^k + \alpha\delta b_1^k u_k^1 \\ &\quad + \alpha\delta a_2^k [a_4^k]^{-1} b_2^k u_k^1 + (1-\alpha)\delta b_1^k u_k^2 \\ &\quad - (1-\alpha)\delta a_2^k [a_4^k]^{-1} b_2^k u_k^2 + O(\delta^2) \end{aligned} \tag{55}$$

completed with the output $y(k)$ which can be written as

$$\begin{aligned} y(k) &= c_1^k - c_2^k [a_4^k]^{-1} \{a_3^k + b_2^k u_{k-1}^2\} + d^k u_k^1 \\ &= c_1^k - c_2^k [a_4^k]^{-1} \{a_3^k + b_2^k u_k^1\} + d^k u_k^1 \end{aligned} \tag{56}$$

where u_k^1 is equal to $\boldsymbol{u}(k)$ and u_k^2 is equal to $\boldsymbol{u}(k+\alpha)$. With this control scheme one has $u^+(k) = u^-(k) = u_k^1 = u_{k-1}^2$.

V.B Sampling of the reduced system with a delayed zero-order hold

As in the previous subsection, we give the system obtained under the RS procedure. Let us consider the nonlinear system, given by (14)-(16), the sampling of the corresponding continuous-time reduced system (23), using the delayed zero-order hold (54), takes the form

$$\begin{aligned} x(k+1) &= x(k) + \delta a_1^k - \delta a_2^k [a_4^k]^{-1} a_3^k + \alpha\delta b_1^k u_k^1 \\ &\quad + \alpha\delta a_2^k [a_4^k]^{-1} b_2^k u_k^1 + (1-\alpha)\delta b_1^k u_k^2 \\ &\quad - (1-\alpha)\delta a_2^k [a_4^k]^{-1} b_2^k u_k^2 + O(\delta^2) \end{aligned} \tag{57}$$

$$y(k) = c_1^k - c_2^k [a_4^k]^{-1} \{a_3^k + b_2^k u_k^1\} + d^k u_k^1 \tag{58}$$

Computing the input output dynamics associated to (56) and (58) gives the same result. This is a consequence of the time scaling of the control scheme (54) delayed with respect to the data acquisitions. More precisely

Proposition V.1 *Under the delayed zero-order hold control scheme (54) verifying the assumption A.4, the input output behaviours associated to the RS and SR procedures are equivalent.*

VI Illustrative examples

To show the efficiency of the proposed control scheme we consider the following illustrative examples.

VI.A A nonlinear example

Let Σ_{ex} be the system

$$\begin{cases} \dot{x} = x^2 + z + u \\ \varepsilon \dot{z} = -z + u \\ y = x - 0.5z \end{cases}$$

with $\varepsilon = 0.0005$ and initial conditions $x(0) = z(0) = 0.5$.
The reduced continuous-time system is

$$\begin{cases} \dot{x} = x^2 + 2u \\ y = x - 0.5u \end{cases} \tag{59}$$

The previous reduced system is not strictly proper although the original system is strictly proper.

Applying the RS scheme to the continuous-time system (59), we find

$$\begin{aligned} x(k+1) &= x(k) + \delta(x^2(k) + 2\boldsymbol{u}(k)) + O(\delta^2) \\ y(k) &= x(k) - 0.5\boldsymbol{u}(k) \end{aligned} \tag{60}$$

The output depends on u so that the relative degree is equal to 0, while the relative degree of the continuous-time system is equal to 1.

Suppose that the desired closed-loop discrete-time output behaviour is

$$y(k+1) = 0 \qquad \forall \ k \in N^* \tag{61}$$

Thus, at time $k\delta$, we get

$$\boldsymbol{u}(k+1) = 2\left[y(k) + 0.5\boldsymbol{u}(k) + \delta(y(k) + 0.5\boldsymbol{u}(k))^2 + 2\delta\boldsymbol{u}(k)\right] \tag{62}$$

with $\boldsymbol{u}(0) = 0$.
As it is shown in the previous section, (60) is not the correct model for digital control. This is underlined below with simulation results. Figure 4 shows that the output behaviour is unstable when the control (62) is applied to the system Σ_{ex}. This is due to the fact that with the fictitious direct link between u and y, we do not take the dynamics of x into account which is unstable after feedback, and we take the measure during the input jump.

Now, let the sampled model with the delayed zero-order hold (54)

$$x(k+1) \quad = \quad x(k) + \delta x^2(k) + 2\alpha\delta u_k^1 + 2(1-\alpha)\delta u_k^2 + O(\delta^2)$$

$$y(k+1) \quad = \quad x(k) + \delta x^2(k) + 2\alpha\delta u_k^1 + 2(1-\alpha)\delta u_k^2 - 0.5u_k^2 + O(\delta^2)$$

so that

$$\begin{array}{ll} u_k^1 = u_{k-1}^2 \qquad \text{with:} \qquad u_0^1 = 0 \\ u_k^2 = \frac{-1}{2(1-\alpha)\delta - 0.5}\left[(y(k) + 0.5u_k^1) + \delta(y(k) + 0.5u_k^1)^2 + 2\alpha\delta u_k^1\right] \end{array} \tag{63}$$

Simulation parameters: $\delta = 0.05\,s$, $\varepsilon = 0.0001$ and $\alpha = 0.5$. The initial state conditions are chosen to be 0.5 and 0.1 for the slow and fast state, respectively.

Figure 5 shows that the output behaviour is stable when the control law (63) is applied to system Σ_{ex}. Consequently, when the output dynamics is assigned and as the zero dynamics is stable, the slow state dynamics is also assigned.

This result confirms the intuitive idea that an input-output sampled representation is "robust" for unmodelled high-frequency dynamics, when a delayed zero-order hold is used, as long as the frequency of the unmodelled dynamics is substantially higher than the sampling frequency (this may be immediately underlined by a simple pole placement).
If we choose in (63) $\alpha = 1$, we find the control law (62), which gives an unstable closed loop. This is due to the fact that assumption A.4 is not verified for this choice of α. Moreover, the output in figure 5 does not converge in one step to zero, this is due to the approximation in the sampled model with the delayed-order hold.

VI.B A linear example

Let us consider the linear system given by ([2]):

$$\begin{aligned} \dot{x} &= x + 2u \\ \varepsilon\dot{z} &= -z + u \\ y &= x + 2z + u \end{aligned}$$

The reduced continuous-time system is

$$\begin{cases} \dot{x} = x + 2u \\ y = x + 2u \end{cases} \tag{64}$$

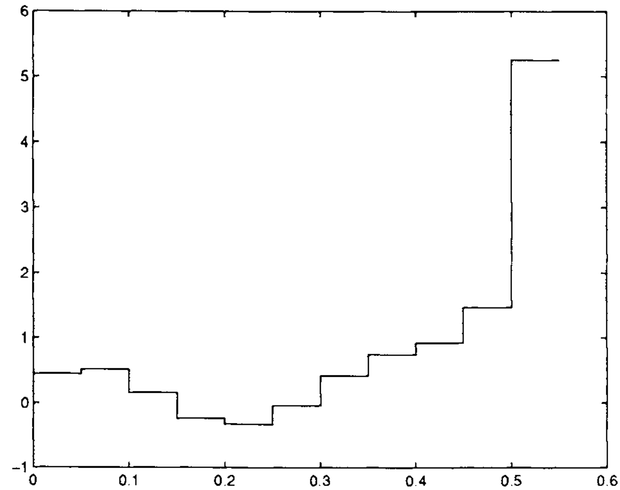


Figure 4 : The measured output y
with the input(62)

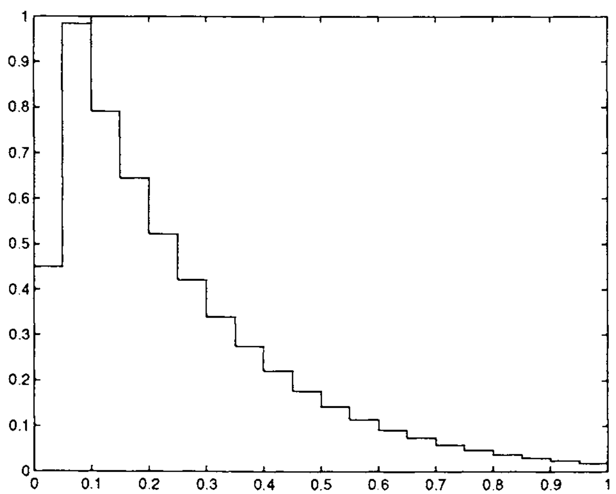


Figure 5 : The measured output y
with the input (63)

In this linear example the full and the reduced system do not have a strictly proper transfer function

$$
\begin{aligned}
G(s,\varepsilon) &= \frac{2}{s-1} + \frac{2}{\varepsilon s + 1} + 1 \\
G(s,0) &= \frac{2}{s-1} + 3
\end{aligned}
$$

Applying the classical discrete-time control (26) to the reduced continuous-time system (64), we obtain

$$
\begin{aligned}
x(k+1) &= e^{\delta} x(k) + 2(e^{\delta} - 1)\boldsymbol{u}(k) \\
y(k) &= x(k) + 2\boldsymbol{u}(k)
\end{aligned}
$$

which gives the discrete-time transfer function

$$
H(z,0) = \frac{3z - (e^{\delta} + 2)}{z - e^{\delta}}
$$

To stabilize this system, we choose the following output feedback

$$
\boldsymbol{u}(k) = \frac{\lambda - e^{\delta}}{e^{\delta} + 2 - 3\lambda} y(k) \tag{65}
$$

where $\lambda = 0.5$ is the desired pole placement in a closed loop.

With the control scheme (54), contrary to the previous example, taking the input sampling $k'\delta = (k+\alpha)\delta$ as reference sampling time one has

$$
\begin{aligned}
x(k'+1) &= e^{\delta} x(k') + 2(e^{\delta} - 1)\boldsymbol{u}(k') \\
\tilde{y}(k') &= e^{\alpha\delta} x(k'-1) + 2(e^{\alpha\delta} - 1)\boldsymbol{u}(k'-1) + 3\boldsymbol{u}(k'-1)
\end{aligned}
$$

where $\tilde{y}(k')$ is the delayed output corresponding to the output at time $((k'-1+\alpha)\delta)$.
We can verify that the discrete-time transfer function with the control (54) is given by

$$
H^{d}(z,0) = \frac{-2e^{\alpha\delta} - e^{\delta} + z(1 + 2e^{\alpha\delta})}{z(z - e^{\delta})}
$$

The stabilization of the considered system with pole placement at $\lambda_1 = 0.9788 + i0.0284$, $\lambda_2 = 0.9788 - i0.0284$ and $\lambda_3 = 0.1650$ gives

$$\boldsymbol{u}(k' + 1) = \frac{-\tilde{y}(k') + b_1 \boldsymbol{u}(k')}{b_2}$$

which can be rewritten in the form of (54), we obtain

$$\begin{array}{ll} u_k^1 = u_{k-1}^2 \qquad \text{with:} \qquad u_0^1 = 0 \\ u_k^2 = \frac{-\tilde{y}(k) + b_1 u_k^1}{b_2} \end{array} \tag{66}$$

where b_1 and b_2 are equal to -21 and 19.6 in order to have desired pole placement.

Simulation parameters are equal to: $\delta = 0.05\,s$ and $\varepsilon = 0.0005$. The initial state conditions are chosen to be 0.5 and 0.1 for the slow and fast state, respectively.

In figure 6 we see that the output behaviour is unstable for the control (65). Contrary to the above performances, satisfactory results are obtained in the second case, when delayed zero-order hold is used with $\alpha = 0.5$. In fact, the fast component has no time to converge to zero, while in the second case this component converges to zero during the second fraction of the sampling period.

Moreover, it is important to note here that there are no errors due to the approximation of the discrete-time model. In fact, the discrete-time model is exact, because the system is linear.

In [3, 28] a particular multi-rate control was used in order to obtain, for a linear singularly perturbed system, the same input-output behaviour for the RS and SR procedures. This was realized by maintaining the input equal to zero during the last step of the multi-rate. The delayed zero-order hold appears to be more suitable in practice (no constraints on the input u).

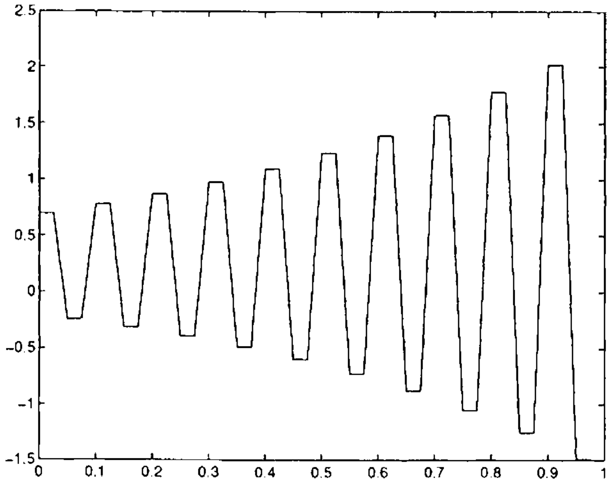


Figure 6 : The measured output y
with the input (65)

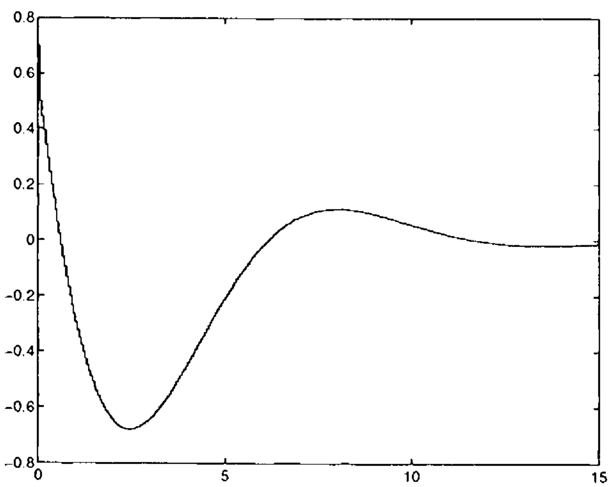


Figure 7 : The measured output y
with the control (66)

VII Conclusion

The problem of modelling a process is of utmost importance in control theory. In fact, for the same process, different models depending on the specific control task may be designed since unmodelled high-frequency dynamics oftently occured as unknown disturbances and parametric uncertainties. The control system must possess sufficient robustness and insensitivity properties. For these reasons, singular perturbations legitimize simplifications of a dynamics model. One of these simplifications is the omission of this high-frequency dynamics which increases the dynamics order of the system. However, a digital control design based on a reduced system may be unsatisfactory even unstable. In this work, we present a digital control scheme which is robust with respect to stable unmodelled high-frequency dynamics. The basic idea of this scheme is to introduce a delay between the input and data measurements. This is done on the basis of the reduced sampled system deduced from a slow sampling. In the first part, some of the basic concepts of nonlinear systems with slow and fast dynamics are introduced followed by a sampled scheme with slow and fast sampling to take into account the time scale property. The improvements with the delayed zero-order hold have been illustrated by two examples.

Appendix

Appendix 1: Notations

- In the text Id represents identity operator, I_d represents the identity function and I represents the identity matrix.

- Consider a formal operator L_X. One defines the exponential formal operator as:

$$e^{\delta L_X} \stackrel{def}{=} \sum_{i=0}^{\infty} \frac{\delta^i}{i!} L_X^i$$

with $L_X^0 = Id$ and $L_X^i = L_X^{i-1}(L_X)$.

- L_X^k represents the composition k times of the operator L_X
- The application of L_X to a function h: $\mathbb{R}^n \to \mathbb{R}$ is defined as

$$L_X\, h = \frac{\partial h(x)}{\partial\, x} X = \sum_{i=1}^{n} \frac{\partial h(x)}{\partial x_i} X_i$$

- The Lie bracket is denoted as

$$ad_F(G) = L_F(G) - L_G(F)$$
$$\text{and} \qquad ad_F^k(G) = ad_F(ad_F^{k-1}(G))$$

- "|" represents the evaluation at a specific point.

Appendix 2: Nonlinear Sampling

Let us consider the nonlinear analytic system of the form:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{aligned} \tag{67}$$

and assume a control variable u constant over intervals of amplitude δ:

$$u(t) := u(k), \quad k\delta \le t < (k+1)\delta, \;\; k \ge 0 \tag{68}$$

Proposition VII.1 *The state and output solutions at time $(k+1)\delta$ of the system (67) under (68), initialized at $x(k)$, are equal to*

$$\begin{aligned} x(k+1) &= F^\delta(x(k), u(k)) \\ &= e^{\delta L_{f(.,u(k))}}(I_d)|_{x(k)} \\ y(k+1) &= e^{\delta L_{f(.,u(k))}}(h)|_{x(k)} \\ &= h(F^\delta(x(k), u(k))) \end{aligned}$$

with $F^\delta : [0, \delta_0[\times M \times \mathbb{R} \to \mathbb{R}^n$ given by

$$\begin{aligned} F^\delta(x(k), u(k)) \;\; = \;\; & e^{\delta(L_f(.,u(k))}(I_d)|_{x(k)} \\ & = x(k) + \delta(L_f(.,u(k))(I_d)|_{x(k)} \\ & + \frac{\delta^2}{2!}(L_f(.,u(k))(L_f(.,u(k))(I_d)|_{x(k)} \\ & + \cdots + \frac{\delta^p}{p!}(L_f(.,u(k))^p(I_d)|_{x(k)} + \cdots \end{aligned}$$

Appendix 3: Baker-Campbell-Hausdorff formula

The following recalls, proposed in the context of nonlinear sampled systems for expanding with respect to u ([19]), are presently used to get expansions with respect to the small parameters ε.

Given two smooth R^n-valued vector fields P and Q: $R^n \to U$, where U is an open set of R^n, e^{L_P} denotes the exponential Lie series given by $e^{L_P} = \sum_{k\geq 0} \frac{1}{k!} L_P^k$ where L_P is the usual Lie derivative operator associated to the vector field P. The following formal series equalities hold:

$$\begin{aligned} E^1(P,Q) &\overset{def}{=} \frac{d(e^{-L_P} e^{L_{P+tQ}})}{dt}_{|t=0} = \frac{(Id - e^{-ad_P})}{ad_P}(Q) \\ \tilde{E}^1(P,Q) &\overset{def}{=} \frac{d(e^{L_{P+tQ}} e^{-L_P})}{dt}_{|t=0} = \frac{(e^{ad_P} - Id)}{ad_P}(Q) \end{aligned} \tag{69}$$

The quotient appearing in the right-hand side of (69) means the formal "cancellation" of the numerator by the denominator.
Generalizing (69), the following operators are iteratively defined

$$\begin{aligned} E^{i+1}(P,Q^{i+1}) &= \frac{d(E^i(P+tQ,Q^i))}{dt}_{|t=0}, \ i \geq 1 \\ \tilde{E}^{i+1}(P,Q^{i+1}) &= \frac{d(\tilde{E}^i(P+tQ,Q^i))}{dt}_{|t=0}, \ i \geq 1 \end{aligned} \tag{70}$$

where Q^i denotes the presence of Q at i-times, in the right-hand side of (70). Expanding the expressions (69), we easily verify the following relations:

$$\begin{aligned} E^1(P,Q) &= \sum_{k\geq 0} \frac{(-1)^{k+1}}{(k+1)!} ad_P^k(Q) \\ \tilde{E}^1(P,Q) &= \sum_{k\geq 0} \frac{1}{(k+1)!} ad_P^k(Q) \end{aligned} \tag{71}$$

which are useful to explicitly compute the vector fields $E^1(P,Q)(I_d)$ and $\tilde{E}^1(P,Q)(I_d)$.
According to these notations, the next theorem can be stated.

Theorem VII.1 *([19]) The following expansion holds true:*

$$e^{L_{P+tQ}} e^{-L_P} = Id +$$

$$\sum_{k\geq 1}\frac{t^k}{k!}\sum_{(i_m+\ldots+i_1)=k}\tilde{C}(i_m,\ldots,i_1)\tilde{E}^{i_m}(P,Q^{i_m})\ldots\tilde{E}^{i_1}(P,Q^{i_1})$$

where the coefficients $\tilde{C}(i_m,\ldots,i_1)$ *are integers, recursively given by*

$$\tilde{C}(i)=1 \qquad \forall i\geq 1$$

$$\begin{aligned}\tilde{C}(i_m,\ldots,i_2,1)&=\tilde{C}(i_m,\ldots,i_2)+\\ &\tilde{C}(i_m,\ldots,i_2-1,1)+\ldots+\tilde{C}(i_m-1,\ldots,i_2,1)\end{aligned}$$

with $\quad \tilde{C}(i_m,\ldots,i_2,i_1)=0 \qquad \forall i_j\leq 0$
Moreover, for all i_1 *such that* $i_1\geq 2$, *we have :*

$$\begin{aligned}\tilde{C}(i_m,\ldots,i_2,i_1)&=\tilde{C}(i_m,\ldots,i_2,i_1-1)+\\ &\tilde{C}(i_m,\ldots,i_2-1,i_1)+\ldots+\tilde{C}(i_m-1,\ldots,i_2,i_1)\end{aligned}$$

Denoting T any R^n valued vector field, a straightforward application of the following well-known equality derived from the Baker-Campbell-Hausdorff formula, that is:

$$e^{ad_P}L_T=e^{L_P}\ L_T\ e^{-L_P}$$

results in the following expression relating $E^1(P,Q)$ and $\tilde{E}^1(P,Q)$:

$$\tilde{E}^1(P,Q)=e^{ad_P}(E^1(P,Q)) \tag{72}$$

Using (72) and theorem VII.1, one can easily prove the following corollary.

Corollary VII.1 *The following expansion holds true:*

$$e^{-L_P}e^{L_{P+tQ}}(I_d)=I_d+\sum_{k\geq 1}\frac{t^k}{k!}\sum_{(i_m+\ldots+i_1)=k}\tilde{C}(i_1,\ldots,i_m)E^{i_m}(P,Q^{i_1})\ldots E^{i_1}(P,Q^{i_m})$$

where the coefficients $\tilde{C}(i_1,\ldots,i_m)$ *are defined in theorem VII.1.*

On the other hand, it can be shown, in terms of integral forms, that:

$$E^1(P,Q) = \frac{e^{-ad_P} - Id}{-ad_P}(Q) = \int_0^1 e^{-\sigma L_P} L_Q e^{\sigma L_P} d\sigma \tag{73}$$

and iteratively

$$E^2(P,Q^2) = \int_0^1 [e^{-\sigma L_P} L_Q e^{\sigma L_P}, \int_0^\sigma e^{-tL_P} L_Q e^{tL_P} dt]\, d\sigma \tag{74}$$

Lemma VII.1 *The following equality holds true:*

$$\begin{aligned} &E^2(P,Q^2) + E^1(P,Q)E^1(P,Q) = \\ &\quad 2\textstyle\int_0^1 \int_0^\sigma e^{-\sigma L_P} L_Q e^{(\sigma - t)L_P} L_Q e^{tL_P} dt d\sigma \end{aligned} \tag{75}$$

Appendix 4: The Gröbner formula

The Gröbner formula enables to compute the solution of analytic equations.

Let us consider the vectorial equation:

$$y = h(w) \qquad \text{with} \quad y \in R^k, \quad \text{and} \quad w \in R^k$$

where h is an analytic function. It is assumed that $(\frac{\partial h}{\partial w})_{|w=w_0}$ is invertible for any w_0 belonging in a neighbourhood of the solution.

Theorem VII.2 *[9] The local inverse function ϕ, solution of the above equation, is given by the following formula*

$$w = \phi(y, w_0) = \{e^{F(y,w_0,.)} I_d\}_{|w=w_0}$$

$$F(y, w_0, .) = \sum_{i=1}^{k} (y^i - h(w_0)^i) L_{D_i}(.)$$

where D_i denotes the i^{th} column of the inverse of the Jacobian matrix of the function h:

$$D \triangleq (\frac{\partial h}{\partial w})^{-1}_{|w=w_0}$$

and y^i and h^i denote the i-component of the vectors y and h, respectively.

The above solution is expressed in terms of Lie serie which converge absolutely and uniformly in a neighbourhoud of the solution. The assumption about invertibility of the Jacobian matrix of the function h at $w = w_0$, is exactly the required assumption in the implicit function theorem. Finally, we note that the initial value w_0 fixes implicitly the validity domain of the solution.

References

[1] J.P. Barbot, "Computer-aided design for sampling a nonlinear analytic system", *in Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Lecture Notes in Computer Science No 357, Springer Verlag, 1989*

[2] J. P. Barbot, N. Pantalos, "Using symbolic calculus for singularly perturbed nonlinear systems", *Lecture Notes in Control and Information Sciences, No 165, Springer Verlag, pp. 49-49, 1991.*

[3] J.P. Barbot and N. Pantalos, "A sampled-data control scheme for stabilization of systems with unmodeled high-frequency dynamics." *IEEE, A.C Vol 37, No 2, pp 285-288, 1992.*

[4] J.P. Barbot, N. Pantalos, S. Monaco and D.Normand-Cyrot, "Discretization schemes for nonlinear singularly perturbed systems." *Proceedings of the 30^{th} IEEE-CDC-91, pp. 443-449, Brighton, 1991.*

[5] J.P. Barbot, N. Pantalos, S. Monaco and D.Normand-Cyrot, "On the control of singularly perturbed nonlinear systems", *Proceedings of NOLCOS 92, BORDEAUX, pp.459-464, 1992*

[6] M. Djemai, J. Hernandez and J.P. Barbot, "Nonlinear control with flux observer for a singularly perturbed induction motor", *Proceedings of IEEE-Conference on Decision and Control CDC-93, St-Antonio, USA, pp. 3391-3396, 1993.*

[7] F. Esfandiari and H.Khalil, "On the robustness of sampled-data control to unmodelled high-frequency dynamics." *IEEE A.C vol 34, pp. 900-903, 1989.*

[8] N. Fenichel, "Geometric singular perturbation theory for ordinary differential equations," *J. Diff. Eq., 31, pp. 53-98, 1979.*

[9] W. Gröbner, "Serie di Lie e Loro Applicazioni", *Poliedro, Gremonese, Roma, 1973.*

[10] J.P. Harcaut, "Nonlinear control of missiles through a geometric approach", *in Lecture Notes in Control and Information Sciences, A. Bensoussan and J.L. Lions, Ed. Vol. 144, pp. 27-36, Springer Verlag, 1990.*

[11] H. Kando, T. Iwazumi. "Multirate digital control design of an optimal regulator via singular perturbation theory", *Int. J. Control, Vol.44, No.6, pp.1555-1578, 1986.*

[12] H.K. Khalil, "Nonlinear Systems," *Mac Millan Ed. 1992.*

[13] H.K. Khalil, "On the robustness of output feedback control methods to modelling errors," *IEEE Trans. on AC, Vol. 26, pp. 524-526, 1981.*

[14] H.K. Khalil, "A further note on the robustness of output feedback control methods to modelling errors," *IEEE Trans. on Automatic Control, Vol. 29, pp. 861-862, 1984.*

[15] K. Khorasani, "A slow manifold approach to linear equivalents of nonlinear singularly perturbed systems," *Automatica, vol.25, No.2, pp. 301-306, 1989.*

[16] P. V. Kokotović, H. K. Khalil and J. O'Reilly, "Singular Perturbation Methods in Control: Analysis and Design", *New York: Academic Press, 1986.*

[17] B. Litkouhi and H. Khalil, "Multirate and composite control of two-time-scale discrete-time systems." *IEEE Trans., AC-30, No.7, pp. 645-651, 1985.*

[18] R. Marino, "Hamiltonian techniques in control of robots arms and power systems", *Theory and Application of Nonlinear Control Systems, C.I. Byrnes and A. Lindquist Ed. pp. 65-73, 1986.*

[19] S. Monaco and D. Normand-Cyrot, "A combinatorial approach of the nonlinear sampling problem." *Lecture Notes in Control and Information Sciences, No.144, 1990.*

[20] S. Monaco and D. Normand-Cyrot, "On the sampling of linear analytic control system," *Proceedings of the 24^{t}h IEEE-CDC-85, Fort Lauderdale, USA, pp. 1467–1462, 1985.*

[21] S. Monaco and D. Normand-Cyrot. "Zero dynamics of sampled nonlinear systems," *Systems & Control Letters, 11: pp. 229–234, 1988.*

[22] S.Monaco, D.Normand-Cyrot, "On the sampling of time delayed nonlinear systems" *Proceddings of European Control Conference ECC-91, Grenoble, pp. 556-559, Ed. Hermes, 1991*

[23] S. Monaco and D. Normand-Cyrot, "A unified representation for nonlinear discrete-time and sampled dynamics." *J. of Mathematical Systems, Estimation and Control, 5, No 1, 1995.*

[24] D. S. Naidu, "Singular perturbation methodology in control systems." *IEE Control Engineering Series 34. 1988*

[25] D. S. Naidu, D. B. Price, J. L Hibey. "Singular perturbations and time scales in discrete control system-An overview." *Proc. of the 26-th IEEE CDC, Los Angeles, USA, 1987.*

[26] D. S. Naidu and A.K. Rao, "Singular Perturbation Analysis of Discrete Control Systems," *Lect. Notes in Math., Springer-Verlag, 1985.*

[27] J. O'Reilly, "Robustness of linear feedback control systems to unmodelled high frequency dynamics." *Int. J. Cont., Vol. 44, pp. 1077-1088, Oct. 1986.*

[28] N.Pantalos,"Systèmes non linéaires regulièrement et singulièrement pertubés: Analyse et commande en temps continu et discret," *thèse Docteur en Sciences de l'Université Paris 11, Orsay 1991.*

[29] G. Peponides, P.V. Kokotović and J.H. Chow, "Singular perurbation and time scales in nolinear models of power systems", *IEEE Trans. CAS, Vol. 29, pp. 758-767, 1982.*

[30] M.W. Spong, "Modelling and control of elastic joints robot", *J. of Dynamics Systems Measurement and Control, Vol 109, pp. 310-319, 1987.*

[31] M. W. Spong, K. Khorasani, P. V. Kokotović, "An integral manifold approach to the feedback control of flexible joints robot," *IEEE J. of Robotics and Automation, RA-3, no 4, 1987.*

[32] V. Saksena, J. O'Reilly, P.V. Kokotović. "Singular Perturbations and Time-Scale Methods in Control Theory: Survey 1976-1983." *Automatica, Vol. 20, No.3, pp. 273-293, 1984.*

[33] A.N. Tikhonov, "Systems of differential equations containing small parameters multiplying some of derivatives" *Mat. Sub. 31, pp. 575-586, 1952.*

[34] M. Vidyasagar, "Robust stabilization of singularly perturbed systems," *Syst. Contr. Lett., Vol. 5, pp. 413-418, May, 1985.*

CAD TECHNIQUES IN CONTROL SYSTEMS

by

M. Jamshidi
CAD Laboratory for Intelligent & Robotic Systems
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, New Mexico 87131

I. INTRODUCTION

The object of this chapter is many folds. After a brief historical note of computer-aided control systems design (CACSD), various techniques and tools will be reviewed. A detail description of several popular CACSD lauguages and packages will be given and a brief survey will follow. Finally, an outlook and some conclusions will end our discussions of the subject. The present discussions is, in part, given by various papers in Jamshidi and Herget [1] and Rimvall and Cellier [2].

Long before computers were used to design a control system, they were being used to <u>simulate</u> them. One possible definition of simulation, according to Korn and Wait [3], is that they represent <u>experimentation with models</u>.Each simulation program consist of two primary segments - one is written to represent the mathematical model of the system, while the other is written to do the experiment itself. One of the earliest simulation programs may be CSMP - Continuous Systems Modeling Program which was a popular simulation environment in the 1960's on various IBM main frame machines. For a long while, CSMP was one of only a few computer programs available to engineers to test dynamic behavior of physical systems. The basic experiments performed in most simulation runs have been to determine the behavior of the system's <u>trajectory</u> under the influence of known signals such us a "step", "sinusoidal" or like. This experiment still constitute the main feature of too many simulation packages of today. In other words, a true simulation program should be capable of investigating all model-related aspects of the system, such as, the errors, uncertainities, tolerances, etc. This is not to say that such programs do not exist. In fact, programs such as IBM's DSL (Dynamic Simulation Language) [4] and ASTAP [5] constitute two notable exceptions. These simulation packages would allow the user to experiment some modeling aspects of physical systems and electronic circuits. Another notable example is the celebrated package SPICE which has been used extensively throughout the world for electronic/electrical network problems. Still, other programs, such us ACSL [6], offer possibilities to linearize and perform steady-state response for a nonlinear system. It is, however, a generally accepted opinion that, while model representation techniques have become more powerful over the past years, relatively less has been done to enhance the capabilities of the simulation experiment description.

The main difficulty with many of those packages has been the data structure offered with it. The scheme the data is structured and handled in many simulation packages are still much the same as they were in 1967, when the CSSL specifications [7] were first introduced. When it comes to CACSD software, the simulation aspect of the system is no longer the main issue. Rather, it is the enhanced experimental description of the system that represents the important issue. In other words, simulation is only one of the many software modules (tools) available within the CACSD software environment. In sequel, we refer to computational algorithms used for specific design purpose as CAD techniques and the software developed to realize them as CAD tools.

However, the truth is that until recently, the CACSD packages' data structures was just as poor as those among simulation languages. The lack of proper data structure would often used to cause a great deal of data and time when going through inflexible question-and-answer protocols of earlier CAD packages. This trend followed until Moler [8] introduced a matrix manipulation laboratory software program he called MATLAB. The only data structure in MATLAB is a double-precision complex matrix. MATLAB offers a set of natural and consistant operators to be performed on matrices. Within MATLAB, a 3 x 3 matrix is defined by

```
A = <3, 4, 5 ; 6, 7, 8 ; 9, 0, 1>
```

or

```
A = <3 4 5 ; 6 7 8
     9 0 1>
```

or

```
A = <3 4 5
     6 7 8
     9 0 1>
```

As seen, elements on the same row are separated by either commas or blanks, and entire rows are separated by either semicolons or a carriage return (CR). The matrices can also be defined by other matrices within the two broken brackets (<···>), i.e.

```
A = <2 * ONES (4,1), EYE (4); <-1 -2 -3 -4 -5>>
```

where ONES (4,1) stands for a 4x1 dimensional matrix of all one's as elements, 2 * ONES (4,1) is, therefore, a 4x1 matrix whose elements are all two's. EYE (4) represents a 4x4 unity matrix which is concatanated from right to the 4x1 matrix of 2's. Therefore, up to the semicolon in the above expression of A we have defined a 4x5 matrix. The last portion of the above expression, a 1x5 row matrix <-1 -2 -3 -4 -5>, is concatanated from below; thus, completing the definition of a 5x5 matrix called A, i.e.

```
A =
      2   1   0   0   0
      2   0   1   0   0
      2   0   0   1   0
      2   0   0   0   1
     -1  -2  -3  -4  -5
```

Assume that one wishes to solve the linear system of equations Ax=b whose solution for a nonsingluar A is $x = A^{-1}b$. In MATLAB, the above system can be solved in two ways:

```
x = INV (a) * b
```

or

```
x = a \ b.
```

The latter expression indicates that b is divided at left by a. While the former represents a full matrix inversion, in the latter expression a Gaussian elimination is performed. As it may already be noted, MATLAB provides a convenient environment for linear algebra and matrix analysis. A tool like MATLAB did exist in early 1970's. It was APL - A Programming Language by IBM. With APL, one could do all the operations that MATLAB can do. However, the biggest difficulty with APL has been its extremely cryptic nature. APL is sometimes called "write only" language. Users of APL needed to think like the exact way that the computer would execute the APL program. Users of MATLAB, on the other hand, would have the computer "think" like the human operator.

It should be noted, however, that MATLAB was not designed for CACSD problems. MATLAB, to quote its creator Cleve Moler, is a "software laboratory for matrix analysis." Nevertheless, when it comes to analysis and synthesis of linear time-invariant systems, MATLAB is exactly what a control engineer would need. To illustrate how MATLAB in its original form, can be used, consider it solving a linear quadratic state regulator problem. Consider a linear [2] time-invariant system

$$\dot{x} = Ax + Bu \tag{1}$$

It is desired to obtain a control which would satisfy the above state equation while minimizing a cost function,

$$J = 1/2 \int_0^\infty (x^T Q x + u^T R u)\,dt \tag{2}$$

where u^T denotes the transpose of vector u. Associated with this problem is the solution of an algebraic matrix Riccati equation (AMRE),

$$0 = A^T K + KA - KSK + Q \tag{3}$$

where $S = BR^{-1}B^T$. The optimal control is given by

$$u^* = -R^{-1} B^T K x \tag{4}$$

One of the earlier solutions of (3) was based on the eigenvalues/eigenvectors of the Hamiltonian matrix [9]

$$H = \begin{bmatrix} A & -S \\ -Q & -A^T \end{bmatrix} \tag{5}$$

This method is incorporated in the following algorithm for the solution of the above LQ problem:

1. Check for system's controllability and stop with a message if system is uncontrollable.
2. Computer the Hamiltonian matrix (5)
3. Compute the 2n eigenvalues and right eigenvectors of H. The eigenvalues will be symmetrical with respect to both real and imaginary axes, as long as the system is controllable.
4. Concatanate those eigenvectors with negative eigenvalues into a reduced 2nxn modal matrix and split them into two nxn upper and lower submatrices, i.e.

$$U = \begin{bmatrix} U_1 \\ \cdots \\ U_2 \end{bmatrix}$$

5. The Riccati matrix is given by $K = \text{Re}\ (U_2^* U_1^{-1})$, where the astrix represents conjugate transpose.

The following MATLAB file (called LQ.mtl) can be written and executed to realize the above algorithm:

```
EXEC ('LQ.mtl')
IF ans <> n, Show ('system not controllable'),RETURN,
END <v,e> = EIG(<a,-b*(r\b') ; -q, -a'>;
e = DIAG(e) ; i=0;
FOR j = 1,2*n, IF e(j) < 0,i = i + 1;v(:,i) =
v(:,j);1:ν(1:ν(1:ν(1:n,1:2ν(n2ν(n2ν(n2i);2i);i);i);i);
k = REAL (u2/ul);
p = - r\b' * k
RETURN
```

where the global variable ans in second line corresponds to the rank of the usual controllability matrix, $Q_c = [B\ AB\ A^2B \ldots A^{n-1}B]$. The above set of codes is clearly very readable once the reader is familiar with the MATLAB commands EIG, DIAG, REAL, etc.

From MATLAB several CACSD packages and languages were derived in a span of 5 years. Table I shows a list of six of the CACSD software programs which have been based on the original MATLAB. Some of these programs are called the "spiritual children" of MATLAB by Cellier and Rimvall [2].

TABLE I. Some MATLAB Driven CACSD Software

Software Name	Location Developed, Year	Principle Developer(s)
1. MATRIXx	Integrated Systems, Inc., 1984	S. Shah, et al
2. CTRL-C	Systems Control, Inc., 1984	J.N. Little et al
3. IMPACT	SWISS Federal Institute of Technology, 1983	M. Rimvall, et al
4. CONTROL.lab	University of New Mexico, 1985	M. Jamshidi, et al
5. PRO-MATLAB	Math Works, Inc., 1985	J.N. Little, et al
6. MATLAB-SC	Philips Research Laboratories, 1985	M. Vanbegin & P.Van Dooren

These CACSD programs will be compared from a number of points of views later on. Parallel to and even before the discovery of MATLAB by the control systems engineering community, a number of non-MATLAB software programs have been created both in North America and Western Europe. A list of eight non-MATLAB CACSD programs are summarized in Table II. It must be noted that there are still a good number of CACSD and simulation packages which have been developed. These software programs are either not relevant to the main theme of this Chapter and are specialized in nature, or are less publicized by their authors. In any event, some of the software programs not listed in Tables I and II will also be briefly discussed later on.

TABLE II. Some non-MATLAB CACSD Software Packages

Software Name	Location Developed, Year	Principle Developer(s)
1. KEDDC (CADACS)	University of Bochum West Germany, 1979	H. Unbehauen & Chr. Schmid
2. LUND	Lund Inst. Technology Lund, Sweden, 1978	K. J. Aström & H. Elmqvist
3. L-A-S	University of Illinois, 1980	S. Bingulac, et al.
4. TIMDOM	University of New Mexico, 1983	M. Jamshidi, et al.
5. CC	California Inst. Technology, 1984	P.M. Thompson
6. TRIP	Belgium 1985	P.P.J. Van den Bosch
7. WCDS	University of Waterloo, 1986	J.D. Aplevich
8. CATPAC	Philips Laboratories Hamburg, W.Germany, 1986	D. Buenz

II. DEVELOPMENT OF CAD METHODS AND PROGRAMS

A. CAD Methods

The development of computer-aided design methods and programs has traditionally been dependent on the theoretical schemes available to the designer. The design methods, by and large, had been graphics in nature and were limited to single-input single-output (SISO) systems described in frequency domain. The major developments in the design of the these control systems were the advents of Nyquist stability criterion, Bode plots, Root locus plots and Nichols charts [10]. During the decades of 1930's through 1960's the use of computers as a design tool was either non-existant or very much limited to writing a subroutine, most likely in FORTRAN, and a main program to call it -

all in a batch mode. For example, in the late sixties or early seventies, finding the eigenvalues of an nxn (n,say 5)matrix would take about half a day, even with reliable eigenvalue/eigenvector routines.

As the control systems became more complex, i.e. with multi-inputs multiple-outputs (MIMO) systems, new design techniques and subsequently CAD tools were needed. Clearly, for MIMO systems, the graphical techniques of SISO systems were not appropriate and could not provide sufficient insight. Kalman, among many others, lead the way to a new domain of system analysis and design - time domain or state-space methods. Here, the system is described by a set of first-order ordinary differential equations. This modern approach seemed more appropriate for many algorithmic design methods. These MIMO design algorithms, e.g. linear quadratic state regulator are applicable equally to both SISO and MIMO systems.

The modern control design methods often resulted in numbers, parameters values, gain factors and so forth which often lacked an "intuitive feel" for what was being achieved by applying these new techniques. However, without adequate insight into the controlled plant, it was often very difficult to determine an adequate controller structure.

Due to these reasons, many control scientists went back to frequency domain in an attempt to find new design methods for MIMO systems. Notable examples of these efforts is the generalization of Nyquist diagram [11], matrix polynomial representation [12], and robust controller design [13]. These techniques, however, involve multiple sweeping and very sensitive to large number crunching.

Another area of development in control system design techniques has been large-scale systems. These systems are associated with high dimensions among other complex attributes such as nonlinearities, delays, etc. The advent of large-scale systems has confronted the control system designer with two new challenges. One is the need for new control philosophies such as hierarchical and decentralized control and the other is the need of special CAD algorithms to handle this class of systems.

An area where relatively little design algorithms exist is nonlinear systems. Most attempts in the design of nonlinear control systems have had to specialize in either narrowing down the class of systems or treated or using linear systems/linearization techniques for otherwise nonlinear problems. The robustness of these refined algorithms are very poor; i.e. they can not easily handle the design of controllers for systems with either unmodeled dynamics or plants whose parameters vary. A prime example of such systems is robot manipulators whose models are highly nonlinear and plant parameters change. A popular scheme to treat nonlinear systems has been adaptive control [14,15] The contoller gains adapt themselves in accordance with the plant parameter variations. Another potential scheme is robust control [16,17] where the controller can handle the plant whose parameters undergo certain variations.

CAD techniques algorithms can be catagorized into three groups: (i) SISO systems techniques, (ii) MIMO systems techniques and (iii) Large-scale systems techniques. Corresponding to this catagorization, as indicated by Cellier and Rimval [2], three classes of algorithms can be distinguished: (i) SISO algorithms which can effectively handle low-order systems, (ii) MIMO algorithms which can efficiently handle high-order systems, and (iii) LSS algorithm which are dedicated for high-order systems. In the theory of linear control systems, most design schemes are based on the systems' canonical form. These forms are based on "minimum parameter data representation". To described this notion, let us consider an nth order SISO system represented by the following transfer function

$$G(s) = \frac{c_{n-1}S^{n-1} + \ldots + c_2 S^2 + c_1 S + c_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_2 s^2 + a_1 s + a_0}$$

where the denominator is known as the characteristic polynomial. In this system representation there are 2n degrees of freedom, i.e. $\{a_0, a_1, \ldots, a_{n-1}, c_0, c_1, \ldots, c_{n-1}\}$. The controllable canonical representation of this system is given by

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 \ldots & .0 \\ 0 & 0 & 0 & 1 \ldots & .0 \\ \ldots & \ldots & \ldots & \ldots & \\ . & & & & \\ . & & & & \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-1} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \ldots \\ \\ \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} c_0 & c_1 & c_2 & \ldots & -c_{n-1} \end{bmatrix} x$$

If the system has n distinct eigenvalues x_i, i = 1,...,n, the Jordan-canonical representation of the system is given by

$$\dot{x} = \begin{bmatrix} \lambda_1 & 0 & 0 & \ldots & 0 \\ 0 & \lambda_2 & 0 & \ldots & 0 \\ 0 & 0 & \lambda_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \lambda_n \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} b_1 & b_2 & b_3 & \ldots & b_n \end{bmatrix} x.$$

The above two representations are only good for SISO (Low-order) algorithms. This indicates that for systems of higher order, the controllable canonical forms are not the most desirable representation. The biggest advantage of this representation is its simplicity. If one is interested in obtaining high-order algorithms, this "simple" representation must be modified by introducing some reducancy among its 2n degrees of freedom. Under this condition, the system would have more than 2n degrees of freedom with linear dependencies existing between them. It is through this redundancy that one can optimize the numerical behavior of controller design algorithms by balancing the parameter's sensitivities [2,18] . More efficient high-order algorithms based on the Hessenberg representation can be obtained [19]. For large-scale systems, as it was mentioned before, one can decompose (reduce) it into a finite number of small-scale subsystems and use lower-order algorthims for each of them. On the other hand, one can use the sparse matrix techniques for the control of the original large-scale system. These techniques keep track of the indices of non-zero elements and are computationally cost effective only for very high-order systems not for low-order or high-order algorithms.

In summarizing the development of CAD techniques algorithms for control systems design, one can classify them into several schemes: SISO vs. MIMO systems vs. LS systems; continuous-time vs. discrete-time; time-domain vs. frequency-domain; linear vs. nonlinear systems, user-friendly vs. non-user-friendly, just to name a few. The last catagory refers to the algorithms in which the user concentrates on the physical parameters of the problem being solved and not the computational or numerical attributes of the algorithms themselves. An example of this situation is a variable-order and variable-step-size numerical integration algorithm in which only the required accuracy - a physical parameter, needs to be specified by the user.

B. CAD Programs

As it was mentioned earlier, up to 1970 no tangible development had taken place for CAD programs used in the design of control systems - multivariable or otherwise. The only available computer programs were some lists of subroutines programs collected by users of a particular line of computers. An example of this was IBM's Scientific

Subroutine Package in the 1960's, which consisted of a score of routines for various problems in linear algebra and numerical mathematics.

In the earlier days, the use of the computer as a design tool for control systems boild down to writing a main program and call one or a set of subroutines. The book by Melsa and Schultz [20] was one of the first published works in this area. In the early to mid 1970's a gradual interest developed both in Europe as well as North America to coordinate efforts to put subroutine libraries together. In fact, colleagues at Swiss Federal Institute of Technology in Zürich lead by Cellier and Rimval [2] helped develope a "program information center" - PIC by the mid 1970's in Europe.

The initial efforts in CACSD programs began in the late 1970's by creating interactive "interface" to existing subroutine libraries such as the one gathered in Europe to reduce the turnaround time. Towards this goal, Agathoklis et al. [21] at the Swiss Federal Institute of Technology came up with the first generation of an interactive CACSD package they called INTOPS. This package proved to be a surprisingly useful new educational tool, but it could not be an effective research tool.

The next step in the development of CACSD turned out to be actually a giant step in the right direction. In the fall of 1980 a conference on numerical and computational aspect of control systems was organized at Lund Institute of Technology at which MATLAB [8] was formally introduced. In a matter of months after that meeting, MATLAB was installed at many locations all over the world. With its on-line "help" facility, MATLAB became a very simple new tool for students of mathematics and system theory. As indicated earlier, MATLAB resembled APL, but without the latter's highly cryptic nature.

In spite of all its fine features, MATLAB was not really designed as CACSD program or tool. In fact, as Cellier and Rimvall [2] noted, from the viewpoint of a control eigneer, MATLAB has the following shortcommings:

1) MATLAB could not handle frequency domain design techniques easily.
2) MATLAB could not handle nonlinear systems.
3) MATLAB did not have any plotting capabilities, which is a major factor in CACSD.
4) MATLAB's programming capabilities through EXEC files are not sufficient. EXEC files cannot be called as functions, only as subroutines. Their input/output capabilities are limited.
5) There is no GOTO statement in MATLAB and, furthermore, WHILE, IF and FOR cannot be properly nested.
6) Using SAVE/LOAD facility of MATLAB would require handling a large number of files, which isdifficult to maintain. A genuine data base would have been desirable.

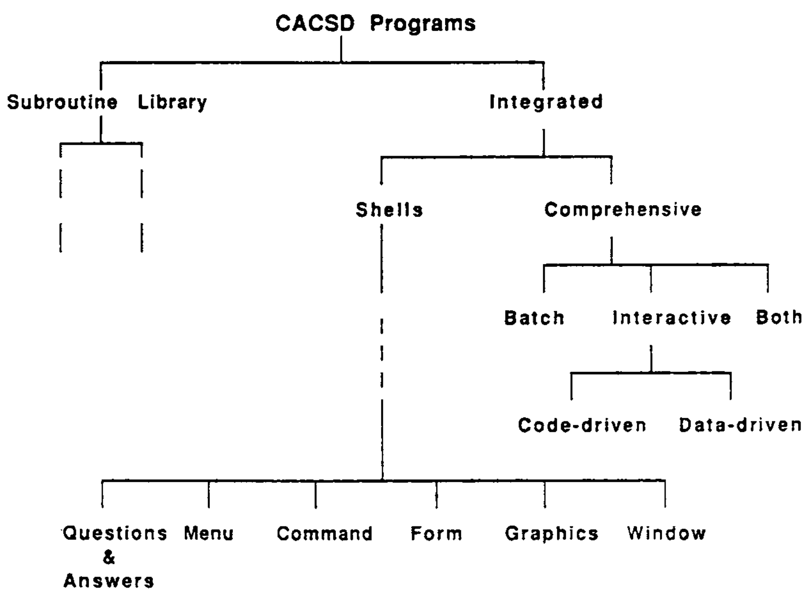


Fig. 1 A classification of CACSD programs.

The above points were, in part, taken into account by a number of subsequent CACSD packages. To summarize the development of CAD programs for control system design, we will go through a classification of CACSD programs. Fig.1 shows one possible classification. As seen, on top, the earlier catagorization of CACSD packages was either a "subroutine library" or an "integrated" set of programs (modules). The early development of these packages were of the former, while the more recent packages are of the latter type. Leaving the earlier "subroutine library" types on the side, the "integrated" CAD packages may be further divided into two types. One is that in which an open-ended operator set of design "shells" would allow the user to code its own algorithm using the CACSD software environment. All the MATLAB - based packages (Section III) are of this type. On the other hand, some packages have gone through a fairly long history of maturity over the years so that we can refer to them as "comprehensive". A very good example of this class of packages is KEDDC (or CADACS) developed in Germany since early 1970's [22,23]. More details on this package is given in Section IV.

From this point, both design "shells" or "comprehensive" packages can be further catagorized in several ways and directions, depending on various input/output data - handling, or actual implementation of the design algorithms. As an example, a "comprehensive" CACSD package can be subdivided into a completely interactive mode, a batch-operated mode, or both. When dealing with nonlinear or even linear time-varying systems, one often has to do a substantial amount of number crunching in "batch" mode. On the other hand, many CACSD packages dealing only with linear time-invariant systems operate in a fully "interactive" mode for a quick analysis and design of such systems. A good example of a package which can be used in both interactive and batch modes is TIMDOM/PC (see Section IV).

Another possible classification of CAD programs for control systems is the style in which algorithms are coded in. This catagorization is either "code-driven" or "data-driven". The packages in which algorithms and operators are implemented as data statements are called "data-driven". These packages are usually written in interpretive languages (e.g. BASIC) and are very useful for experimentation. However, they are not terribly efficient in execution. On the other hand, "code-driven" packages are based on compiled programs whose algorithms and operators are implemented in program code. The algorithms and operators of "code-driven" packages are usually much faster in execution and represent a more stabilized state of a package's development.

A final possible catagorization of CACSD packages is through the styles of "driving through" it, i.e. moving on from the beginning to the end of a given CAD session. As it is shown on the botton of Figure 1, six possible style can be mentioned. In the "question and answer" style, the user is asked numerous questions to determine the flow of the program. This style of user interface, although very simple to implement, is not useful for research-oriented problem solving. "Menu-driven" packages are those in which the user is alway given a choice list of various categories of CACSD problems. The user can then choose one of the choices on the list and that specific program is loaded and made available to the user. The user can make its choice by either a pointer on the screen or a mouse. Many non-MATLAB packages available today are menu-driven (see Section IV). Some believe that this type of user interface shares too much information, leading to a slower package. However, we believe that carefully designed menu-driven interfaces would be more educational then research use.

The other popular interface is "command-driven". Here, the user would need to have the initiative to figure out what to do next. The CACSD program would "prompt" the screen with a symbol to indicate to the user that is ready for the next "command". All MATLAB-based (Section III) packages are command-driven". One disadvantage of this style of interface is that the user has to remember what commands are available and how they work. The HELP facility in MATLAB-based programs would come in handy here.

A "form-driven" interface is most useful during the set-up stage of a problem solution. The screen is often divided into separate alphanumeric fields, each representing one parameter of the algorithm. By moving over fields, the user can override default parameter values. This type of interface is heavily dependent on the hardware and very few CACSD programs utilize it.

The early utility of "graphic-driven" interface in CACSD applications was to display the system's time and frequency responses. Being highly dependent on the terminal and hardware, graphics still remains the most difficult issue in any CACSD package. The basic approach in any alleviating this problem has been to utilize a graphics library which supports a wide variety of terminals drivers to put between the package and terminal itself. However, these libraries have been too experience, and their full utility has been questionable. The recent acceptance of the "graphic kernel system" - GKS as standard [24] is clearly a positive step in the right direction toward eliminating hardware dependencies on terminals. As indicated by Cellier and Rimvall [2], fancy graphics often call for high-speed communication links, which would drive costs up again. GKS standardization would, of course, help drive cost down. The appearance of special-purpose graphics workstations such as APOLLO Domain and SUN would provide high-speed enhanced graphics capabilities at reasonable prices.

Another graphics-related outcome in CACSD package developments has been the graphics-input feature of some programs. Examples of this are SYSTEM-BUILD capability of MATRIXx [25] and BUILD features of CC [26] and PRO-MATLAB [27]. A powerful upcoming CACSD package for nonlinear systems, suggested by Robinson and Kisner [28] uses artificial intelligence to develope a truly graphics-driven package. In this package, as well as in MATRIXx, control system blocks are automatically translated by a graphics compiler into a coded model format.

A last interface is that of "window-driven" variety (see Figure 1). Here, the screen is divided into a number of logical windows, each representing one logical unit--similar to the way physical devices in a computer system used to be. Within each window, one may employ any one of the above described interfaces, i.e. question-and-answer, menu-driven, command-driven, or form-driven. In a recently developed robot simulation package, ROBOT_S by O'Neill and Jamshidi [29], a number of windows are utilized (on a SUN workstation), each with its own menu-driven interface. Another CACSD package which allows mixture of two interfaces is IMPACT [30], a MATLAB-based package, which has extended MATLAB's HELP facility by an extensive QUERY facility. Using QUERY, the user can receive "help" at either command level or at an entire session level. In this way, the user can choose an almost surely question-and-answer extreme all the way to an solely command-driven feature on IMPACT. More details can be obtained in a recent survey by Cellier and Rimvall [2]. Further advanced topics and development on CACSD will be forthcoming in another volume by Jamshidi and Herget [31].

III. MATLAB-BASED CACSD PACKAGES

As it has been discussed earlier, there are at least 7 CACSD packages which has been designated as the "spiritual children" of MATLAB (see Table I), including proMatlab - a modern version of MATLAB. In this section an attempt is made to highlight the common and distinguished features of some of these packages. Due to the lack of availability of all MATLAB-based packages, only four of these packages will be covered in a brief manner. These are CTRL-C [32], MATRIXx [25], CONTROL.lab [33] and PC-Matlab [27].

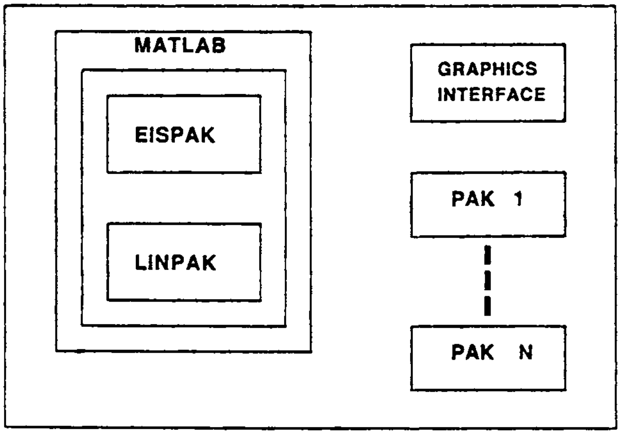


Fig. 2. A generic schematic of a typical MATLAB-based CACSD package.

The advent and impact of MATLAB on the CACSD environment in the 1980's has already been brought up in some detail. Furthermore, asides from MATLAB's convenient matrix data structure, from a numerical analysis point of view, numerically stable algorithms have been used throughout this "laboratory" of matrix analysis which have stemmed from previous successful packages such as EISPAK [34] and LINPAK [35]. Figure 2 shows a generic structure of a typical MATLAB-based package. These package typically have a graphics interface to either externally-supported plotting language or its own internal plotting package, and a number of other modules which we have, generically, designated as PAK1, PAK2,..., PAKN. Therefore, the common features of all these packages are MATLAB'S EISPAK and LINPAK algorithms, and HELP facility, as well as their command-driven nature. For this reason, the linear algebra capabilities are very much the same.

However, each of these packages do possess some subtle differences which we will try to point out in the following subsections.

A. CTRL_C

One of the earliest MATLAB-based programs which came into existance in 1984, was CTRL_C by Systems

Control Technology, Inc. the principle authors were J. Little, A. Emami-Naeini, and S. N. Bangert [32]. In CTRL_C, as in all MATLAB-based packages, all variables are stored in a large stack which resides in either semiconductor or virtual memory, or both. There is, however, a natural data communication between this stack and disk files. All MATLAB-based packages inform the user of readiness to accept the next command. In CTRL_C, the combination of "[>" appears on the screen, while in MATRIXx and CONTROL.lab, MATLAB's original facing broken brackets "<>" are used. In PC-Matlab, the symbols ">>" are utilized to prompt the user.

1. Matrix Analysis

To enter a matrix in CTRL_C, in all such packages, a simple list is used inside two brackets whose rows are distinguished by a semicolon ";". For example, the input line

```
[> a = [1 5 9 3; 2 6 1 4; 3 7 1 6; 4 8 2 8]
```

in CTRL_C would result in the output

```
A =
     1.   5.   9.   3.
     2.   6.   1.   4.
     3.   7.   1.   6.
     4.   8.   2.   8.
[>
```

From this point on, this 4x4 matrix, denoted by A, will be stored for later use in the same terminal session. The matrix analysis in CTRL_C, steming from MATLAB, is very simple. For example, the relation (or command) b = a' would result in the transpose of matrix A defined above. Matrix multiplication, inversion, determinant, rank, condition number, etc. can similiarly be determined by

```
[>   c = a * b;
[>   ain = inv (a);
[>   d = det (a);
[>   r = rank (a);
[>   cn = cond (a);
```

where the semicolon at the end of each statement, as in all MATLAB-based programs, signals to the package not to print out the results, but rather save them in appropriate newly-defined variables. The original MATLAB [8] came with 77 commands, key words or symbols such as ":", FOR, DO, IF, etc. From those, 16 were purely dedicated linear algebra (matrix analysis) commands. Some of the common matrix functions in CTRL_C, inherited from MATLAB, are

```
eig (a)   - eigenvalue and eigenvectors
exp (a)   - matrix exponential
inv (a)   - inverse
svd (a)   - singular-value decomposition
Schur (a) - Schur decomposition
```

However, commands such as geig (a,b), which finds the generalized eigenvalues and eigenvectors of *a*, is a dedicated CTRL_C command. Polynomials are represented by a vector of its coefficients ordered by their descending power. In CTRL_C, conv (a,b) can be used to find the product of polynomials a and b, i.e.

```
[> a = [1 2 1]; b = [1 2];
[> c = conv (a,b)
```

yields:

```
c =
     1.   4.   5.   2.
```

Several other desireable operations such as polynomial division, root finding, characteristic polynomial, and other polynomial operations are similarly supported.

2. Engineering Graphics

In CTRL_C, like many well-developed MATLAB-based packages, a score of graphics commands and features such "Plot", "log-log", "3d Plots", "labels", "title", "axes", etc. are supported. As an example, the following four CTRL_C commands would creat a simple sine function:

```
[> t = 0 : .05 : 4 * pi; y = sin (t);
[> Plot (t,y), title ('sin (t)')
```

which is shown in Figure 3. The first statement would define a vector consisting of elements from 0 to 4π in increments of 0.05. The second statement (on first line) would create a vector of sine values of time vector t. The third statement would plot vector y vs. vector t and provide a title for it.

CTRL_C would allow one to use 3D plots for a better understanding of the structure of large system matrices. For example, "to look" at a 59th order aircraft system matrix, one can use command P3d(a) to produce a 3-dimensional plot of Figure 4, where the height h represents the value of an element above the X-Y plane. This plot would give a perspective to the designs which would not be evident by looking at 3600 numbers.

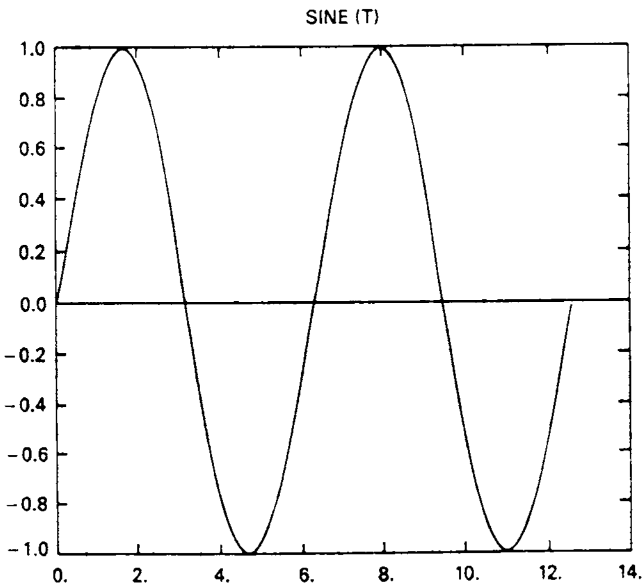


Fig. 3. A simple sine function plotted in CTRL-C

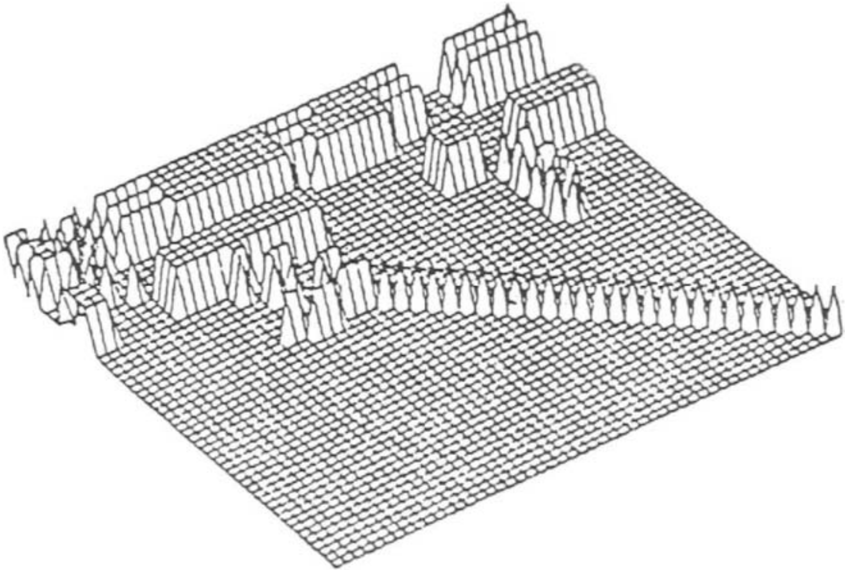


Fig. 4. A 3-dimensional representation of a matrix

3. Control Systems

It is clear that for linear time-invariant systems in state-space form, matrix environments are particularly useful. Within CTRL_C systems may be represented in discrete-time, continuous-time, in polynomial notation, as a Laplace transfer function, or a z-transform transfer function. Within CTRL_C, one can move from time-domain to frequency-domain and vice versa. In sequel, a simple example illustrates some of the features of CTRL_C.

Example 1: Consider a 3rd order SISO system as described in Fig.5. In this example various features of CTRL_C are illustrated. To begin, the numerator and denominator coefficients for the first block are entered:

```
[> num = [1  2]; den1 = [1  .4  1];
```

To determine the open-loop poles of the first block,

```
[>  dr = root (den)
DR =
       -0.2000 + 0.9798i
       -0.2000 - 0.9798i
```

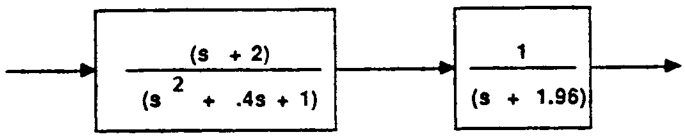


Fig. 5. System of Example 1.

The natural frequency and damping ratio of this block are easily determined by

```
[>  wn = abs (dr)
    WN =
          1.0000
          1.0000

[>  zeta = cos (imag (log (dr)))
     ZETA =
          0.2000
          0.2000
```

To cascade the second block of the system in Fig. 5, a new vector of denominator of this block is introduced,

```
[>  den2 = [1  1.96];
```

Through the convolution command, the series connection denominator can be obtained:

```
[>  den = conv(den1,den2)

   DEN =
          1.0000     2.3600     1.7840     1.9600
```

This equivalent cascaded system can be realized in state space form by the following statement:

```
[> [a,b,c,d] = tf2ss (num,den)

    D =
          0
    C =
          0.    1.    2.
    B =
          1.
          0.
          0.
    A =
          -2.3600    -1.7840    -1.9600
           1.0000     0.0000     0.0000
           0.0000     1.0000     0.0000
```

With the system now in state space format, one can determine time and frequency responses. To do that in CTRL_C, as in all other MATLAB-based packages, using colon ":" command, one can first define a time base, i.e. the statement

```
[> t = 0 : · 1 : 10;
```

creates a vector of 101 time points spanning from 0 to 10 in steps of 0.1 or 100 ms. The following two statements would compute the impulse and step responses of the system using the newly created time base:

```
[> yi = impulse (a,b,c,1,t);
[> ys = step (a,b,c,d,1,t);
```

The 101x1 vectors yi and ys represent the system's output time responses under the influence of an impulse and a step, respectively. To plot these responses, one can simply type in,

```
[> Plot (t,yi,t,ys)
```

which results in the plots of Fig. 6.

In a similar fashion, one can plot frequency responses. This is achieved by using the function

LOGSPACE to create a vector of frequency points equally spaced between two decades, i.e.

```
[> w = logspace (-1,1);
[> [mag, phas] = bode (a,b,c,d,1,w);
```

where matrices mag and phas contain the magnitude and phase responses at the frequencies in vector w. The magnitude and phase are plotted on log.log scales and titled in the upper right corner of the screen with the commands

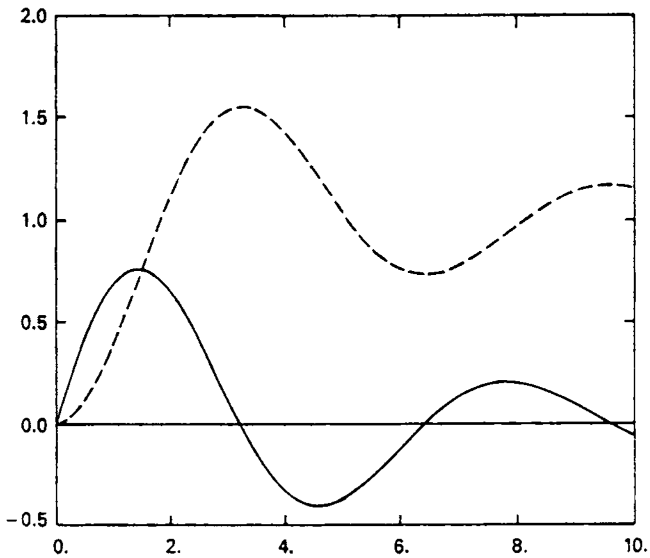


Fig. 6. Time responses for system of Example 1.

```
[> window ('222')
[> Plot (w,mag, 'loglog')
[> title ('magnitude')
```

Similar commands be used to plot the phase and Nichols chart, for example, as shown in Fig. 7.

Finally, one can use pole placement algorithm of Ackerman [13] coded in command PLACE to design a state feedback for this open-loop system. Assume that the desired poles are -3, -3 $\pm$ j3 which are stored away in a vector P.

Then the feedback gain k is obtained by:

```
[> p = 3 * [-1 ; (-1 + i) ; (-1-i)];
[> k = Place (a,b,p)
   K =
        6.6400    34.2160   52.0400
```

which can be checked to provide the desired poles by:

```
[> e = eig (a - b * k)
   E =
        -3.0000 + 3.0000i
        -3.0000 - 3.0000i
        -3.0000 + 0.0000i
```

which does check.

The reference feedforward matrix M is calculated to provide unity DC gain,

```
[> m = 1/(d - (c - d * k)/(a - b * k) * b)
   M =
        27.0000
```

To determine the impulse and step responses of the closed-loop system, the following statements would perform that:

```
[> Ac = a - b * k; Bc = b * m; Cc = Dc = d*m; d - d*k;
[> yi = impulse (ac,bc,cc,1,t);
[> ys = step (ac,bc,cc,dc,1,t);
[> Plot (t,yi,t,ys)
```

which results in Fig. 8.

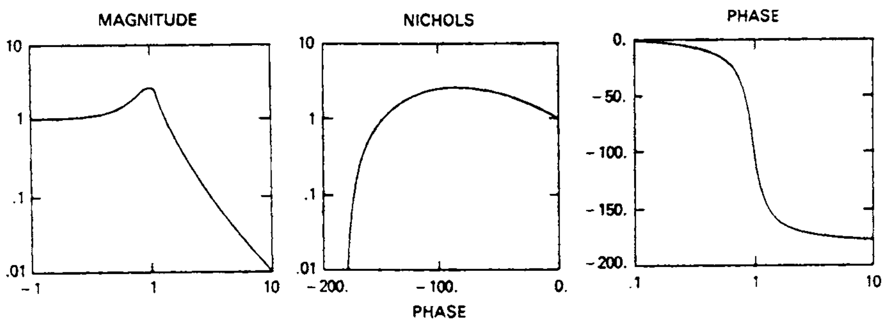


Fig. 7 Frequency responses for system of Example 1.

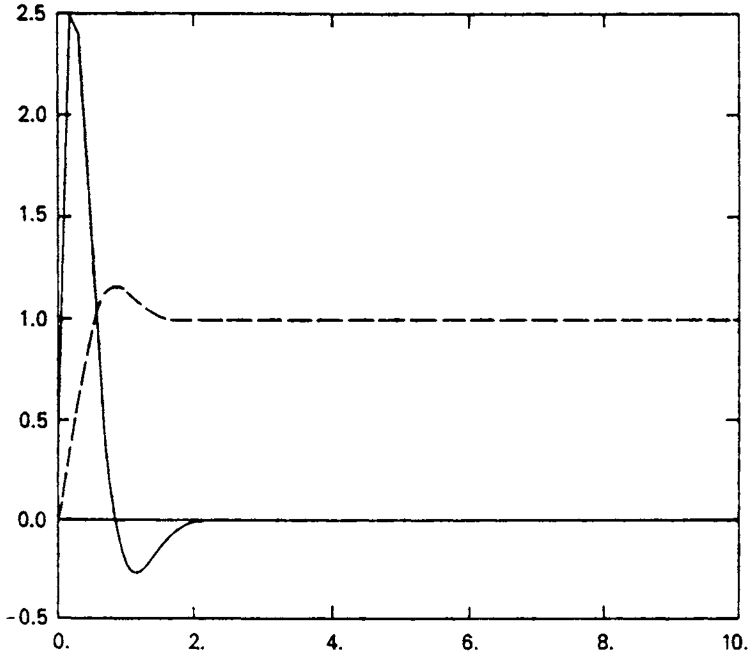


Fig.8. Time responses for the closed-loop system of Example 1.

CTRL_C has numerous other capabilities that space does not allow us to cover. In the area of signal processing, for example, CTRL_C has fast Fourier transform, filter design, etc. Moreover, within CTRL_C, the user can use its "procedure" capability to create new commands. This capability is available on all MATLAB-based packages, and discussion of this type will be provided later.

B. MATRIXx

MATRIXx is generally considered the first commercially available CACSD package based on MATLAB. The structure of this package can be summarized through the following broad catagories:

a) Matrix, vector and scalar operations
b) Graphics
c) Control design
d) System identification and signal processing
e) Interactive model building (SYSTEM BUILD)
f) Simulation and evaluation

The structure of MATRIXx is shown in Fig. 9. We can summarize the capabilities of MATRIXx (Version 6.0, May, 1986) through Tables III - VIII show a summary of the available tools. As it can be seen from these Tables the capabilities of MATRIXx are quite vast. Covering the full capabilities of MATRIXx would be a book in itself. Instead of going into the details of this CACSD package, we will introduce a number of examples to illustrate its features.

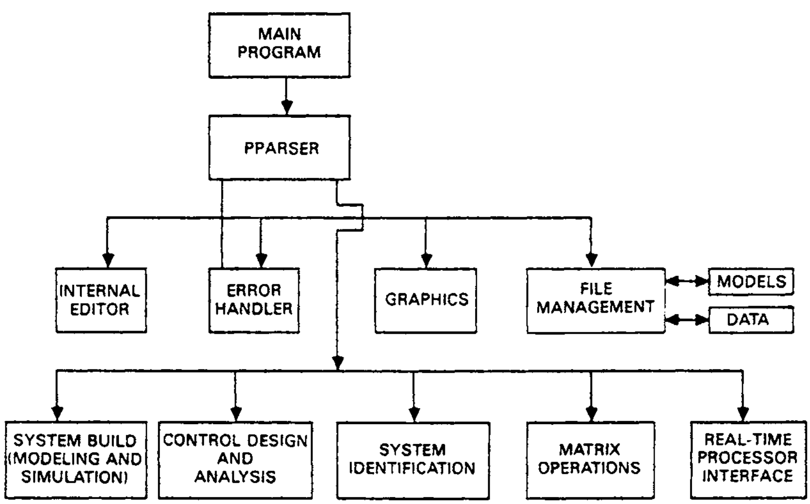


Fig. 9 Structure of MATRIXx

TABLE III. Examples of MATRIXx Capabilities: Matrix Arithmetic.

Data Entry, Display and Editing
Addition, Subtraction, Multiplication and Division
Absolute Value, Real Part, Imaginary Part and Complex Conjugate of a Matrix
Sum and Product of Matrix Elements
Element-by-Element Multiply and Divide
SIN, COS, ATAN, SORT, LOG, EXP of Matrix Elements
Eigenvalue, Singular Value, Principal Values, Schur, LU, Cholesky, OR, Orthogonal, Hessenberg, and Reduced row echelon Decomposition of Matrices
Random Vector and Matrix Generation and Manipulation
Eigenvectors, Generalized eigenvectors, Pseudo inverse, condition number, rank, norm, inverse Hilbert matrices, and rational numbers

---

Table IV. MATRIXx Capabilities: Graphics.

Flexible Commands
Multiple Plots
Axis Labels and Plot Title
Symbols, Lines and Styles
TIcs and Grids
Log Scales
Bar Charts
Labeling and Scales
Polar Plots
Plot Location and Size
Personalization
Report Quality
3-D Graphics
- Parallel & Perspective Projections
- Surfaces
- Curves
- Viewing Transformation

Font size, style and changes
Handcopy, redrawing

Example 2: The definition of a surface is a family of points defined by vectors X and Y, and a matrix Z so that Z is dim(x) by dim(y). The surface is then $z=f(x,y)$. Individual points are linearly connected.

As an example, consider the following MATRIXx statements:

```
<> X = [-2*PI:.35:2*PI]';Y = X;
<> Z = SIN(X)./X*(SIN(Y)./Y)';
<> PLOT(X,Y,Z)
```

which produces the surface plot of Fig. 10.

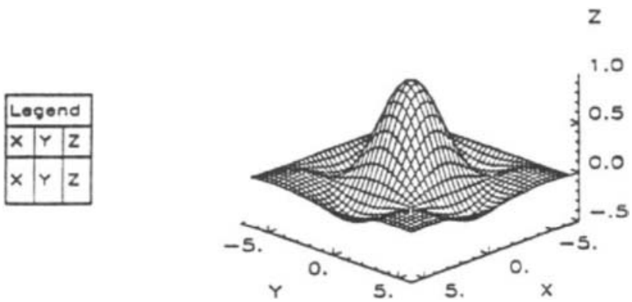


Fig. 10. A three-dimensional plot produced by MATRIXx.

TABLE V. MATRIXx Capabilities: System Description and Simulations.

System Description
Continuous-time Systems
Transfer-Function Descriptions
State-Space Desciptions
Conversion Between State-Space and Transfer-Function Forms
Discrete-Time Systems
Transfer-Function Descriptions
State-Space Descriptions
Conversions from Continuous-Time to Discrete-Time Descriptions
Systems in Modal Form

Appending System Dynamics
Series Connections
Parallel Connections
General Interconnections
System Simulation
Continuous-time Systems
Impulse Response
Step Response
Initial-value Response
Response to General Inputs
Discrete-time Systems
Pulse Response
Step Response
Initial-value Response
Response to General Inputs
Nonlinear Simulations
Simulate general nonlinear continuous models
Simulate general nonlinear discrete models
Simulate general nonlinear multirate models
Simulate general nonlinear hybrid multirate models

TABLE 6. MATRIXx Capabilities: Control Design and System Analysis Capabilities (Applicable to Continuous,Discrete and Hybrid Systems)

Classical Tools
Root Locus
Bode Plots
Nyquist Plots
Nichols Plots
Gain and Phase Margins
Modern Tools
Pole Placement
Optimal Control Design, Discrete and Continuous
Optimal Filter Design, Discrete and Continuous
Frequency-Shaped LQG Design
Singular-Value Decomposition of the Return-Difference
Eigensystem Decomposition Including the Jordan Canonical Form
Model Following Control
Linearization of Nonlinear Systems
Minimal Realization and Kalman Decomposition

Geometric Control Algorithms
Multivariable Nyquist Plots
Transmission Zeros
Algebraic Riccati Solutions
Model Transformation and Reduction
Controllability, Observability and Minimal Realization
    Control Realization
    Observable Realization
    Minimal Realization
    Stair-Case Form
    Eigenvalue Test
    Grammian Test
Transforming to Internally Balanced Form
Transforming to Modal Form
    Evaluating Modal Residues
Model Reduction

In MATRIXx, a MIMO linear time-invariant systemm $\dot{x} = Ax + Bu$, $y = Cx + Du$ is represented by a system matrix

$$S = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

which can be realized by

<> S = [a, b; c, d];

On the other hand, to extract the system's matrices, one can use command "SPLIT":

```
[A,B,C,D] = SPLIT (S,NS);
```

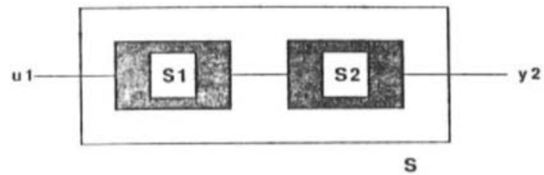
or

```
A = SPLIT (S,NS);
```

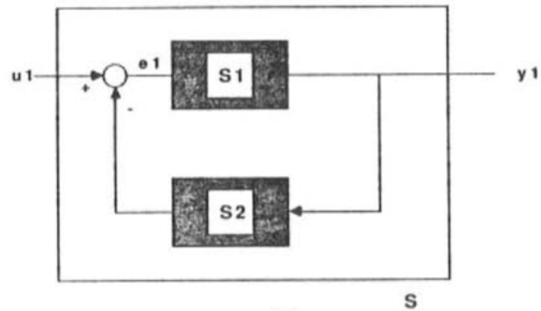
where NS is a variable representing the number of states. The same format is used for discrete-time systems.

One of the more powerful set of commands in MATRIXx are tools for connecting linear systems in series, parallel, or feedback configurations: SERIES, PARALLEL, FEEDBACK, AFEEDBACK, APPEND, and CONNECT commands. A sequence of these commands can reduce connected systems to a single equivalent system. This is useful for input-output control design, state-space control design, and system response analysis.

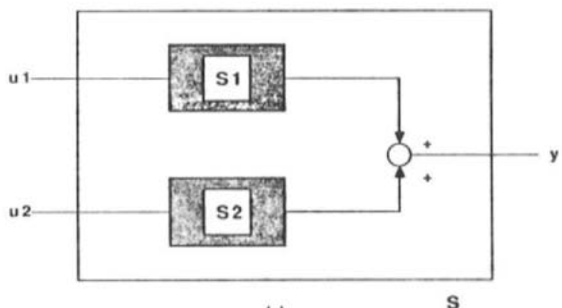
COMMAND	FORMAT	DIAGRAM	FUNCTION
SERIES	[S,NS] = SERIES(S1,NS1,S2,NS2)	(a)	$S = \begin{pmatrix} A1 & 0 & B1 \\ B2*C1 & A2 & B2*D1 \\ D2*C1 & C2 & D2*D1 \end{pmatrix}$
PARALLEL	[S,NS] = PARALLEL(S1,NS1,S2,NS2)	(b)	$S = \begin{pmatrix} A1 & 0 & B1 & 0 \\ 0 & A2 & 0 & B2 \\ C1 & C2 & D1 & D2 \end{pmatrix}$
FEEDBACK	[S,NS] = FEEDBACK(S1,NS1,S2,NS2) [S,NS] = FEEDBACK(S1,NS1,S2) [S,N2] = FEEDBACK(S1,NS1) 1. General 2. constant-gain feedback S2 3. unity feedback.	(c)	$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ y_1 \end{pmatrix} = (S) \begin{pmatrix} x_1 \\ x_2 \\ u_1 \end{pmatrix}$



(a)



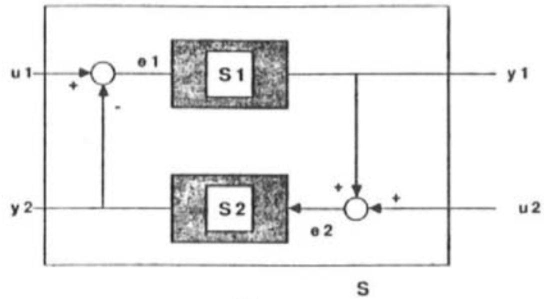
(b)



(c)

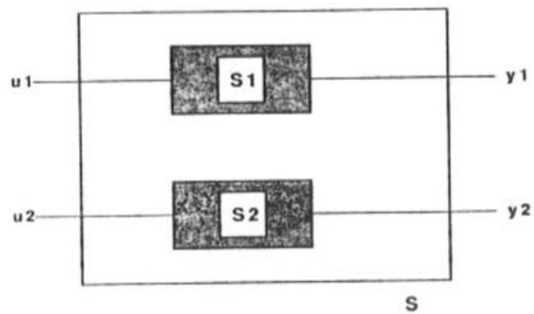
Figure 11 | MATRIX x's block connection commands for multivariable systems.

Figure 11 *(continued)*

COMMAND	FORMAT	DIAGRAM	FUNCTION
AFEEDBACK	[S,NS] = AFEEDBACK(S1,NS1,S2,NS2) [S,NS] = AFEEDBACK(S1,NS1,S2) [S,NS] = AFEEDBACK(S1,NS1) 1. General 2. constant-gain feedback (S2) 3. unity feedback		$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\\ y_1\\ y_2\end{pmatrix} = (S)\begin{pmatrix}x_1\\ x_2\\ u_1\\ u_2\end{pmatrix}$ (d) Feedback—augmented input/output.

(d)

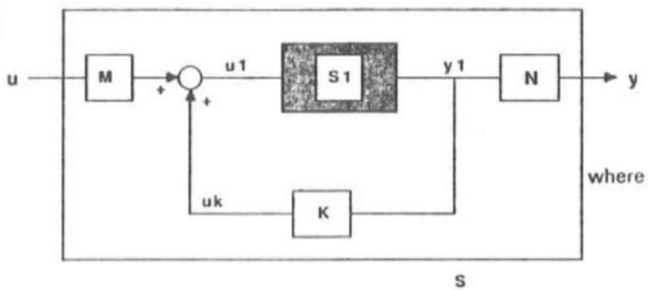
The AFEEDBACK command proceeds a closed-loop configuration with augmented inputs and outputs as shown in this diagram. It is identical to the FEEDBACK command in both format and function.

COMMAND	FORMAT	DIAGRAM	FUNCTION
APPEND	[S,NS] = APPEND(S1,NS1,S2,NS2) (e) Parallel—append of two systems.		$S = \begin{pmatrix}A1 & 0 & B1 & 0\\ 0 & A2 & 0 & B2\\ C1 & 0 & D1 & 0\\ 0 & C2 & 0 & D2\end{pmatrix}$

(e)

The APPEND command appends two state-space systems [S1,NS1] and [S2,NS2] in a form suitable for interconnecting with the CONNECT command. A number of systems can be appended by appending two at a time.

Figure 11 *(continued)*

COMMAND	FORMAT	DIAGRAM	FUNCTION
CONNECT	[S,NS] = CONNECT(S1,NS1,K,M,N) [S,NS] = CONNECT(S1,NS1,K,M) [S,NS] = CONNECT(S1,NS1,K) 1. General 2. unity output gain 3. unity input and output gains		$S = \begin{pmatrix} A1 + B1 * V * K * C1 & B1 * V * M \\ N * W * C1 & N * D1 * V * M \end{pmatrix}$ where $V = (I - K * D1)^{-1}$ $W = (I - D1 * K)^{-1}$

(f)

(f) Feedback with unity gains.

Fig. 11 shows a brief summary of these commands. These commands can be used in succession for a complex system, and once it is reduced to a simple form prior to analysis, simulation or design. Consider the following example:

Example 3: Consider a complex system as in Fig. 12.

```
   Consider a complex system:

<> S1 = [-1 1;1 0];NS1 = 1              Define SISO system 1 = 1/(s+1)
<> S2 = [0 1 0;-1 -0.2 1;1 0 0;0 1 0];NS2 = 2;
                                        Define MIMO system 2 (Oscillator - 1
                                        input, 2 outputs)
<> [S,NS] = APPEND(S1,NS1,S2,NS2);      Append systems 1 and 2
<> FEEDBK = [-1 0 0;0 -3 -.2];          Define feedback gain matrix to double
                                        frequencies.
<> [S,NS] = CONNECT(S,NS,FEEDBK,IGAIN): Connect FEEDBACK around system.
<> [A,B,C,D] = SPLIT(S,NS);             Extract new system components.
```

The next example illustrates the use of LSIM to obtain time response of a system to a general input. The basic format is given by:

[T,Y] = LSIM (S,NS,U,DELTAT,XO)

where U is the input array to be provided by the user, XO is the initial condition vector, DELTAT is the time increment between points in the U array. T and Y are the time vector and output response matrix, i.e. if the system has r outputs, then Y is described by:

$$Y = \begin{pmatrix} y1 & y2 & \ldots & yr \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{array}{l} 1^{st} \text{ time point} \\ \vdots \\ \text{last time point} \end{array}$$

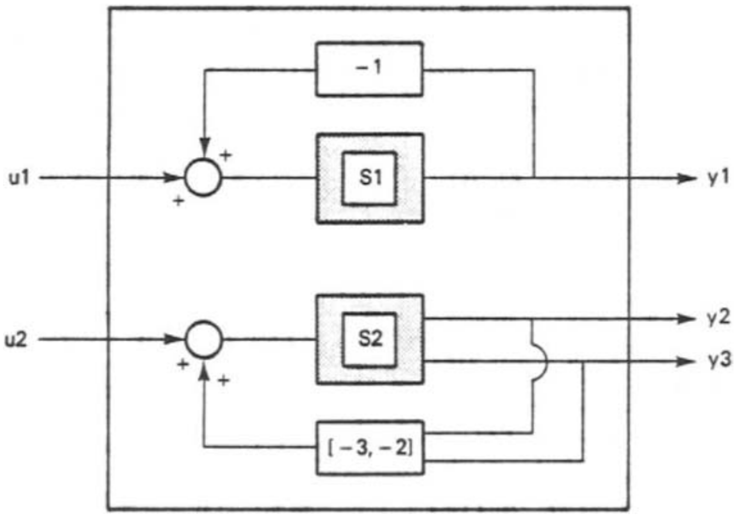


Fig. 12 Block diagram of system of CAD Example 3.

TABLE VII. MATRIXx Capabilities: System Indentification, Signal Processing and Data Anaylsis Capabilities.

Data Display
Time-History Plots
Multichannel Cross-Plots
Scatter Plots
Frequency Plots
Histograms
Data Transformation and Spectral Analysis
Multiplexing/Demultiplexing
Detrending
Censoring
Digital Filtering
Discrete Fourier Transform
Inverse Fourier Transform
Autocorrelation
Cross Correlation
Autospectrum
Cross Spectrum
Decimation and Interpolation
Maximum Entropy Spectrum Estimation

System Identification
Step-Wise Regression and Model Building
Maximum Likelihood Identification of State-Space Models and Nonlinear Models (generated by SYSTEM_BUILD)
Recursive Maximum Likelihood Identification
Extended Kalman Filter Algorithm

Filter Design
Window-Based Methods
Wiener Filter
REMEZ Exchange Algorithm for Finite Impulse Response Filters
Elliptic, Chebyshev, Butterworth Infinite Impulse Response Design

TABLE VIII. INTEGRATION ALGORITHMS AVAILABLE IN MATRIXx

Euler	-	Euler	
Rk2	-	Runge-Kutta	(2nd order)
Rk4	-	Runge-Kutta	(4th order)
Kutta-Merson		(fixed step)	
Kutta-Merson		(variable step)	
DASSL	-	implicit stiff predictor-corrector	

Example 4:

```
<> S = [0,1,0;-1,-.2,1;1,0,0];NS = 2;    Define second order system.
<> RAND('NORMAL');
<> U = RAND(100,1);                      Generate white noise input.
<> DT = 0.2;
<> [T,Y] = LSIM(S,NS,U,DT);              Calculate response to input U.
<> PLOT(T,[U Y]);                        Plot the input and the output.
```

Figure 13 shows the plots of input and output.

The next example utilizes the inverted pendulum problem of CAD Example 5 and place its closed-loop poles using the command POLEPLACE for SISO systems. Example 5: <> // Pole placement for an Inverted pendulum problem

<> // Kwakernaak, H. and Sivan, R. "Linear Optimal Control Systems". Wiley, New York, 1972.

```
<> A = [0 1 0 0;0 -1 0 0;0 0 0 1;-11.65 0 11.65 0];
<> B = [0 1 0 0]';
<> // Desered poles are -5 ± j8.66 and // -8.66 ± j5
<> POLES = [-5+8.66*JAY,-8.66+5*JAY];
<> KC = POLEPLACE (A,B,POLES)
KC     =
     1.0D+03 *
          0.3848     0.0263     -1.2431    -0.2618
```

Command	Format	Diagram	Function
SERIES	[S,NS] = SERIES(S1,NS1,S2,NS2)		$S=\begin{pmatrix} A1 & 0 & B1 \\ B2*C1 & A2 & B2*D1 \\ D2*C1 & C2 & D2*D1 \end{pmatrix}$
PARALLEL	[S,NS] = PARALLEL(S1,NS1,S2,NS2)		$S=\begin{pmatrix} A1 & 0 & B1 & 0 \\ 0 & A2 & 0 & B2 \\ C1 & C2 & D1 & D2 \end{pmatrix}$
FEEDBACK	[S,NS] = FEEDBACK(S1,NS1,S2,NS2) [S,NS] = FEEDBACK(S1,NS1,S2) [S,N2] = FEEDBACK(S1,NS1)		

1. General

2. constant-gain feedback S2

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ y_1 \end{pmatrix} = (S) \begin{pmatrix} x_1 \\ x_2 \\ u_1 \end{pmatrix}$$

3. unity feedback

AFEEDBACK [S,NS] = AFEEDBACK(S1,NS1,S2,NS2)
[S,NS] = AFEEDBACK(S1,NS1,S2)
[S,NS] = AFEEDBACK(S1,NS1)

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ y_1 \\ y_2 \end{pmatrix} = (S) \begin{pmatrix} x_1 \\ x_2 \\ u_1 \\ u_2 \end{pmatrix}$$

1. General
2. constant-gain feedback (S2)
3. unity feedback

The AFEEDBACK command procides a closed-loop configuration with augmented inputs and outputs as shown in the diagram above. It is identical to the FEEDBACK command in both format and function.

APPEND [S,NS] = APPEND(S1,NS1,S2,NS2) $S = \begin{pmatrix} A1 & 0B1 & 0 \\ & 0A2 & 0B2 \\ C1 & 0D1 & 0 \\ & 0C2 & 0D2 \end{pmatrix}$

(e) Parallel-append of two systems.

The APPEND command appends two state-space systems [S1,NS1] and [S2,NS2] in a form suitable for interconnecting with the CONNECT command. A number of systems can be appended by appending two at a time.

CONNECT [S,NS] = CONNECT(S1,NS1,K,M,N

[S,NS] = CONNECT(S1,NS1,K,M)

[S,NS] = CONNECT(S1,NS1,K)

$$S = \begin{pmatrix} A1+B1 * V *K*C1 & B1*V*M \\ N* W * C1 & N *D1*V*M \end{pmatrix}$$

$$\text{where } V = (I - K * D1)^{-1} \quad W = (I - D1 * K)^{-1}$$

1. General
2. unity output gain
3. unity input and output gains

(f) Feedback with unity gains.

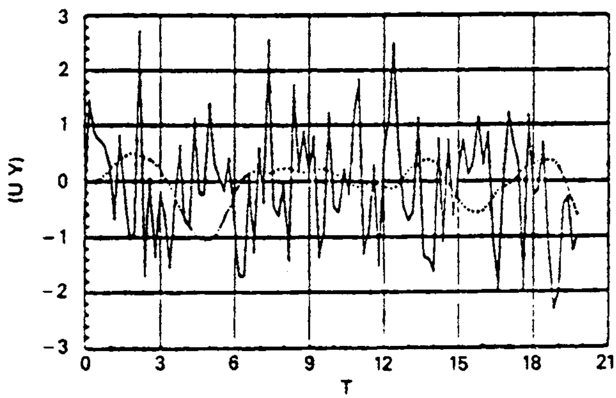


Fig. 13. Input/output response for Example 4.

Within MATRIXx, the user can program its own code via four programming tools. These are (a) macros, (b) command files, (c) user routines, and (d) functions. Each scheme has its advantages and disadvantages from a programming efficiency point of view. Each scheme will be briefly defined with more details on functions and followed by an example.

Macros. A macro is a sequence of MATRIXx operations which can be executed with a single command. A macro is stored as text within MATRIXx. Note that macros should be saved once they have been created, say through MATRIXx's editor, before leaving MATRIXx.

Command File. Command files contain sequences of MATRIXx commands that are usually longer than those used in macros. Command files are saved as data files and can be used in different sessions. More important, command files can be accessed by many users, while macros are defined only for the user who defined it.

User Routines. In MATRIXx, as in its original predecessor, MATLAB, a "user" subroutine can be written in FORTRAN to perform a specific task which is not supported by MATRIXx. *Functions.* Function have three main advantages for the first two schemes:

1. Functions allow parameter passing, hence can be used like any other MATRIXx command.
2. Functions use local variable; that is, a function defined by a user can be utilized by another by simply changing the local variables.
3. Functions will remain an integral part of MATRIXx, MATRIXx just like ".m" files in PRO-MATLAB.
   The general form of MATRIXx function is given by
   //[output1,output2,...] = FUNNAME
   (Input1,Input2,...)
   List of MATRIXx Commands
   RETF

The example below uses the MATRIXx' function which was created to generate an N x N Hilbert matrix.

Example 6:

```
//A = HILBM(N)

For I = 1 : N,. . .
    A(I,J) = 1/(I + J - 1);
RETF
.
.
.
<>N = 3;
<>b = HILBM(N);
<>b
B =

    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
```

Before leaving our discussion of MATRIXx, we remind readers that MATRIXx constitutes one of the more mature CAD environments for control systems. The full details on it can best be obtained by consulting the manual and/or actually [36] using it on a terminal.

C. CONTROL.lab

CONTROL.lab is another MATLAB-based CACSD package which was developed at CAD Laboratory for Intelligent and Robotic Systems, University of New Mexico, by Jamshidi et al. [33,37]. In many respects CONTROL.lab is similar to CTRL-C and MATRIXx; therefore, the language will be briefly described and a few CAD examples will be given instead.

CONTROL.lab is an interactive computer-aided language that serves as a convenient medium for computations involving linear multivariable systems and in some cases nonlinear systems. Aside from the matirx analysis capabilities of CONTROL.lab, its system analysis capabilities range from standard stability tests (Routh, Jury, Lyapunov, etc.) analytical solutions of linear systems (e^{AT}, A^{K}, $(sI-A)^{-1}$, $C(sI-A)^{-1}B+D$), etc), and controllability/observability test tests, to simulation and time response of continuous-time and discrete-time systems. The design capabilities of CONTROL.lab range from standard pole placement schemes through state and output feedback through P, PD, PI and PID, from solutions to algebraic and differential matrix Riccati equations to the optimal linear quadratic problems. One of the stronger capabilities of CONTROL.lab is in the area of estimation and filtering. In this area, several primitives exist to design Kalman filters and states estimators (observers).

The package has a dedicated graphics interface to Tektronix's PLOT 10 IGL [36] and allows the use of more popular terminals such DEC's VT240 and Tektronix's TEK 4010, to name two. A pictorial catagorization of CONTROL.lab is shown in Fig.14. These various classes of primitives are described in its User's Guide [36].

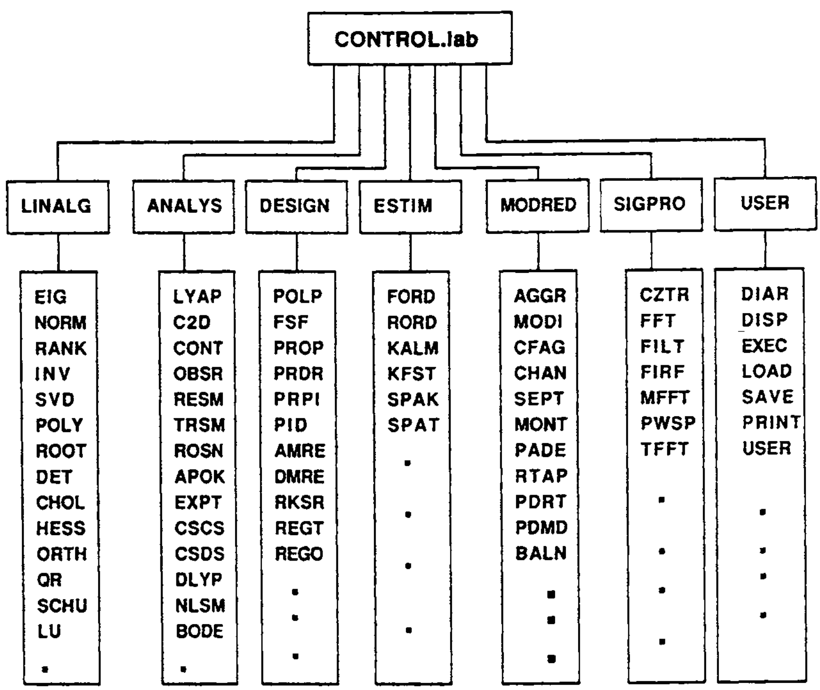


Fig. 14. A pictorial categorization of CONTROL.lab.

Example 7: In this example, we consider the numerical inversion of a transfer function.

$$X(s) = \frac{b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots a_1 s + a_0}$$

where a_i and b_i are real constant coefficients and n is a positive integer. The differential equation corresponding to the above transfer function is

$$x^{(n)}(t) + a_{n-1}x^{(n-1)}(t) + \cdots + a_1\dot{x}(t) + a_0 x(t) = 0$$

with initial conditions:

$$\begin{aligned}
x(0) &= b_{n-1} \\
\dot{x}(0) &= b_{n-2} - a_{n-1}x(0) \\
\ddot{x}(0) &= b_{n-3} - a_{n-1}\dot{x}(0) - a_{n-2}x(0) \qquad (1) \\
&\vdots \\
x^{(n-1)}(0) &= b_0 - a_{n-1}x^{(n-2)}(0) - \cdots - a_1 x(0)
\end{aligned}$$

In state form, the transfer function can be written by defining

$x = (x_1\ x_2\ \cdots\ x_n)\ T = (x\ \dot{x}\ \ddot{x}\ \cdots\ x^{(n-1)})T$, i.e.,

$$\dot{x} = Ax = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ & & & & & 0 \\ 0 & & & & & 1 \\ -a_o & -a_1 & \cdots & \cdots & \cdots & -a_{n-1} \end{bmatrix}$$

with initial state x(0) given by Eq.(1). As an example, consider a third order transfer function,

$$X(s) = \frac{s^2 + s}{s^3 + 5s^2 + 5.25s + 5}$$

with initial state:

$$x_1(0) = x(0) = b_{n-1} = b_2 = 1$$
$$x_2(0) = \dot{x}(0) = b_1 - a_2 x(0) = -4$$
$$x_3(0) = \ddot{x}(0) = b_0 - a_2 \dot{x}(0) - a_1 x(0) = 14.75$$

and state form

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -5.25 & -5 \end{pmatrix} x \qquad (2)$$

To find the inverse Laplace transform of (2) on CONTROL.lab, one can use CSCS for the following system,

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -5.25 & -5 \end{pmatrix}, \ B = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \ C = (1\ 0\ 0),\ D = (0).$$

The CONTROL.lab session is given by:

```
A = <0  1  0;  0  0  1;  -5  -5.25  -5>;
<> b = < 0 0 0 >;
<> c = < 1 0 0 >;
<> d = 0;
<> e = < 0 5 .1 .001 >;// initial time, final time,
   step size, and tolerance
<> xo = <1 4 14.75>';
<> CSCS (a,b,c,d,xo);
```

The resulting function x(t) vs. time is given in Figure 15.

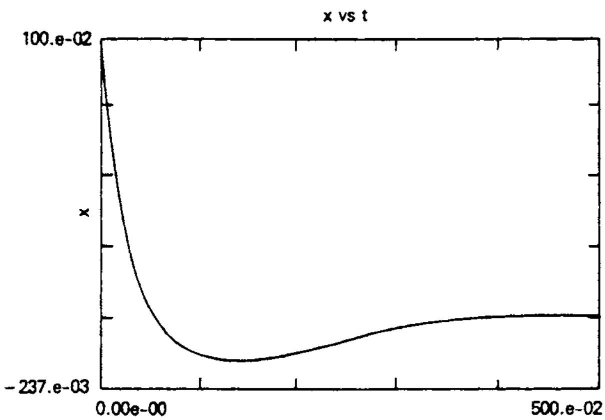


Figure 15. Inverse Laplace transform function of Example 7.

Example 8: In the example, a nonlinear system, shown in Fig. 16 is used to simulat using CONTROL.lab. For the system shown, let r(t) = 3.5 u(t), select final time to be 4.0 with a step size of 0.08 second and obtain the print-plot of the output c(t).

Let the nonlinear block be represented by N and proceed to find an observable companion form for the system such that the desired output is one of the state variables instead of a combination of several state variables. Example 9: In this example, a nonlinear system, shown in Fig. 16, is used to simulate using CONTROL.lab. For the system shown, let r(t) = 3.5 u(t), select final time to be 4.0 with a step size of 0.08 second and obtain the pring-plot of the output c(t).

Let the nonlinear block be represented by N and proceed to find an observable companion form for the system such that the desired output is one of state variables instead of a combination of several state variables.

$$\frac{C(s)}{R(s)} = \frac{\dfrac{6N}{s(s+2)\ (s+10)}}{\dfrac{6N}{s(s+2)\ (s+10)}} = \frac{6N}{s(s+2)\ (s+10) + 6N}$$

$$= \frac{6N}{s^3 + 12s^2 + 20s + 6N}$$

$$C(s)(s^3 + 12s^2 + 20s + 6N) = R(s)(6N)$$

$$\dddot{c} + 12\ddot{c} + 20\dot{c} + 6Nc = 6Nr$$
$$\dddot{c} = 6Nr - 6Nr - 12\ddot{c} - 20\dot{c}$$

Integrating three times and ordering the terms:

$$c = \int (-12c + \int (-20c + \int (6Nr - 6Nc))) \, dtdtdt$$

The simulation diagram is shown in Figure 17. From this figure, it follows that:

$$\dot{x}_1 = 6Nr - 6Nx_3$$
$$\dot{x}_2 = x_1 - 20x_3$$
$$\dot{x}_3 = x_2 - 12x_3$$

$$\therefore \quad A = \begin{bmatrix} 0 & 0 & -6N \\ 1 & 0 & -20 \\ 0 & 1 & -12 \end{bmatrix}, \quad B = \begin{bmatrix} 6N \\ 0 \\ 0 \end{bmatrix}, \quad C = [0 \quad 0 \quad 1]$$

Using the primitive "STEQ", the observable companion state equation derived above can be checked.

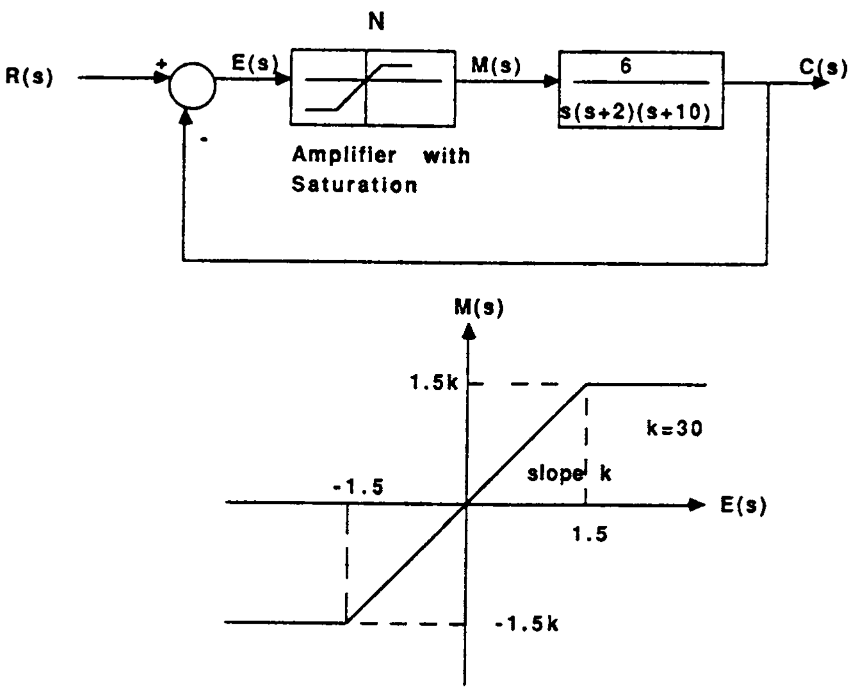


Fig. 16. A nonlinear system for Example 8.

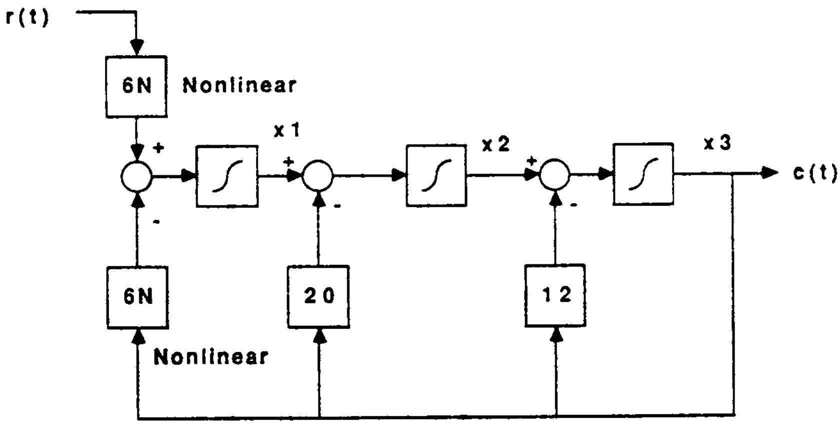


Fig. 17. Simulation diagram of a nonlinear system for Example 8.

Note: "99" is entered in place of the variable "6N".

```
<> p = <99  20  12  1>;
<> q = <99  0  0  0>;
<> r = /</ 3  2>;
<> <a,b,c> = steq(p,q,r)
C =
     0  0  1
B =
     99
      0
      0
A =
     0  0   -99
     1  0   -20
     0  1   -12
```

Next the routine describing the system is entered:

```
% cat nl.f
SUBROUTINE G(T,Y,YP)
DIMENSION Y(3), YP(3)
DOUBLE PRECISION T,GAIN,ERROR,R,Y,YP
R = 3.5D0
ERROR = R - Y(3)
IF (ERROR .LT. -1.5) THEN
```

```
     GAIN = -45.0D0
ELSE IF (ERROR .GT. 1.5)THEN
     GAIN = 45.0D0
ENDIF
YP(1) = 6.0D0 * GAIN
YP(2) = Y(1) - 20.0D0 * Y(3)
YP(3) = Y(2) - 12.0D0 * Y(3)
RETURN
END
SUBROUTINE G1
RETURN
END
SUBROUTINE G2
RETURN
END
```

Then the subroutines are compliled and linked with CONTROLAB:

```
% f 77 -o controlab nl.f - lcontr lplot10
```

Then CONTROLAB is called:

```
% controlab
```

Enter the system order, initial conditions, and solution range and step size:

```
<> n = 3;
<> y = <0  0  0>;
<> indx = 0  4.0  0.08>;
```

Finally, solve the system using the nonlinear simulation primitive "NLSM:"

```
<> <time,vars> = nlsm(n,h,index)
     TIME =          VARS =
          0.0800     0.          0.          0.
          0.1600     21.6000     0.8563      0.0183
          0.2400     43.2000     3.3525      0.1187
          0.3200     64.8000     12.6028     0.3304
          0.4000     86.4000     12.6028     0.6553
             .          .           .           .
             .          .           .           .
             .          .           .           .
          3.7600      87.6844    31.4213     2.4579
          3.8400     101.3566    34.9213     2.6579
          3.9200     111.5737    39.0002     2.9334
          4.0000     117.4516    43.2453      3.2554
```

A quick plot of c(t) can be obtained by noting that c(t) = var(i,3), i = 1,2,$\cdots$.

Hence, use a do loop

```
FOR I = 1:51, CT(I) = VARS(1,3);
```

then Plot(time,ct) would give a quick plot shown in Fig.18. A more illustrative plot can be obtained using "GPLT".

CONTROL.lab is only available on a VAX environment under VMS. Its biggest shortcoming is the lack of macro editing and extendibility beyond what the original MATLAB can do. Efforts are underway, however, to refine the feature of this CACSD software program.

D. PC_MATLAB

PC_MATLAB represents a complete reprogrammed enhanced version of the original MATLAB written in "C" language. This task was achieved by Little and Moler [27] of Math Works, Inc. This new software program is designed in a very modular fashion, somewhat even more convenient than extensive packages such as CTRL_C or MATRIXx. It is, therefore, an optimized, second generation MATLAB for MS-DOS personal computers. Its most useful features, in our opinion, are its programmable macros and the fact that most of its macros are transparent to the users as so-called ".m" files.

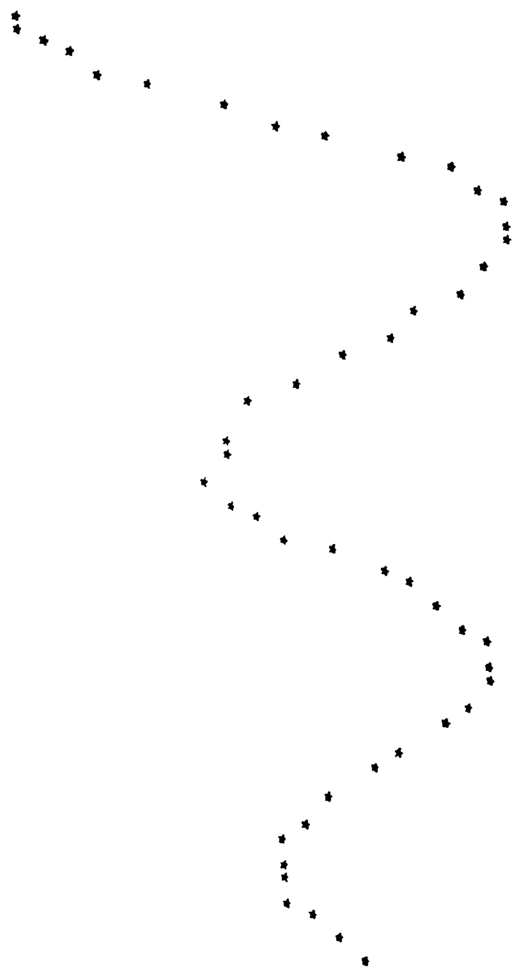


Fig. 18. A graphical plot for Example 8, ct vs. time.

PC_MATLAB can be augmented with several other so-called "tool boxes" in such areas as control, signal processing, identification, etc. These tool boxes can be obtained afterwards and "integrated" into a more comprehensive CACSD package. Table IX shows a summary of control tool box of PC_MATLAB. It is noted that this MATLAB-based package has essentially all the features of larger CACSD packages. In addition to these commands, PC_MATLAB has four other commands which are helpful in model building process. These are:

APPEND	Append the dynamics of two subsystems
CONNECT	System interconnection
PARALLEL	Form the parallel connection of two systems
SERIES	Cascade two system in series

TABLE IX. Primitives of PC_MATLAB's Control Tool Box.

	MODELING AND CONVERSIONS
ss2tf	state-space to trnasfer function conversion
ss2zp	state-space to zero-pole conversion
tf2ss	transfer function to state-space conversion
tf2zp	transfer function to zero-pole conversion
zp2tf	zero-pole to transfer function conversion
zp2ss	zero-pole to state-space conversion
c2d	continuous to discrete-time conversion
d2c	discrete to continuous-time conversion
append	append system dynamics
connect	block diagram modeling
parallel	parallel system connection
series	series system connection
	TIME RESPONSE
impulse	impulse response
step	step response
lsim	continuous simulation to arbitrary inputs
dimpulse	discrete-time unit sample response
dstep	discrete-time step response
dlsim	discrete simulation to arbitrary inputs
filter	SISO z-transform simulation
	GAIN SELECTION
lqr	linear-quadratic regulator design
lqe	linear-quadratic estimator design
dlqr	discrete linear-quadratic regulator design
dlqe	discrete linear-quadratic estimator design
place	pole placement
rlocus	root-locus
	FREQUENCY RESPONSE
bode	Bode plots
nyquist	Nyquist plots
dbode	discrete Bode plots
freq	SISO z-transform frequency response

UTILITY

```
damp        damping factors and natural frequencies
margin      gain and phase margins
ctrb        controllability matrix
obsv        observability matrix
tzero       transmission zeros
fixphase    unwrap phase for Bode plots
ord2        generate A,B,C,D for a second-order system
ric         continuous Riccati equation residuals
dric        discrete Riccati equation residuals
abcdcheck   check the consistency of an (A,B,C,D) set
nargcheck   check number of .m file arguments
```

which can be used to simulate systems described in block diagram form.

A few examples follow to further illustrate the features of PC_MATLAB.

<u>Example</u> <u>9</u>: In this example, two discrete-time systems will be simulated in PC_MATLAB. The first is a 3rd order system whose step response is sought. The second is a 2nd order system whose response to a white noise is determined.

```
>> % First System Simulation
>> Phi = [ 0 1 1 ; -.5 -.5 1 ; 0 0 - .5 ];
>> gama = [ .5 .5 ; 0 .05; .05 0 ];
>> c = [ 1 0 0 ; 0 1 0 ]; d
>> yd1 = dstep (Phi, gama, c,d,1,20);
>> yd2 = dstep (Phi, gama, c,d,2,20);
>> kt = [ 0 : 1 : 19 ];
>> Plot (kt, yd1, kt, yd2)
>> title ('Step Responses - Discrete - Time Systems'),
   xlable ('Time')
>> ylabel ('y1 & y2')
>> % Second System Simulation
>> a = [ 0 1 ; -1 -1 ]; b = [ 0 ; 1 ] c = [ 1 0 ];
>> d = 0 ; xo = [ 0 ; 0 ];
>> % Form an arbitrary random input
>> rand ('normal')
>> u = rand (50,1);
>> yn = dlsim (a,b,c,d,u,xo);
>> Plot (yn)
>> title ('White Noise Response 2nd Order System')
>> xlabel ('Time'), ylable ('y(k)')
```

Fig. 19 shows the resulting responses created above.

Example 10: In this example, the "EXEC" file 'LQ.mtl' presented in Section I will be re-examined within the framework of PC_MATLAB. As mentioned before, PC_MATLAB would come with numerous ".m" files which utilize the capabilities of the software to solve a new problem. Below is a listing of PC_MATLAB's "lqr.m" file, which effectively performs the same task as "LQ.mtl" file.

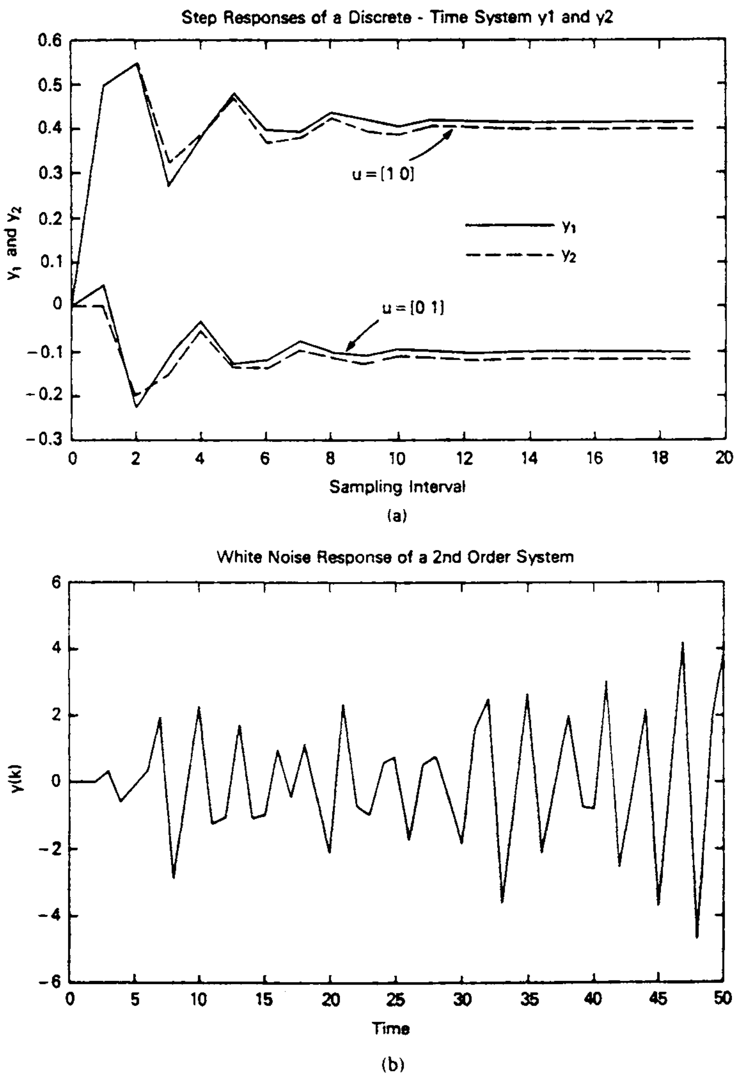


Fig. 19. Two discrete-time systems responses, for Example 9.

(a) Step responses of a discrete-time system y_1 and y_2.

(b) White noise response second-order system.

```
function [k,s]=lqr(a,b,q,r)
%LQR   Linear quadratic regulator design for
       continuous-time systems.
%      [K,S] = LQR(A,B,Q,R) calculates the optimal
       feedback gain matrix K
%      such that the feedback law:
%
%              u = Kx
%
%      minimizes the cost function:
%
%              J = Integral {x'Qx + u'Ru}dt
%
%      subject to the constraint equation:
%                .
%                x = Ax + Bu
%
%      Also returned is S, the steady-state solution
       to the  associated
%      algebraic Riccati equation:
%                 0 = SA + A'S - SBR^-1  B'S + Q
if  nargcheck(4,4,nargin)
       return
end
if  abcdcheck(a,b)
       return
end
[m,n] = size(a);
[mb,nb] = size(b);
[mq,nq] = size(q);
if  (m ~= mq)  |  (n ~= nq)
     errmsg('A and Q must be the same size')
     return
end
[mr,nr] = size(r);
if (mr ~= nr)  |  (nb ~= mr)
          errmsg('B and R must be consistent')
          return
end
```

```
% Check if q and r are positive definite
chol (q);
chol (r);
[v,d] = eig([a b/r*b';q,-a']); % eigenvectors of
   Hamiltonian
d = diag(d);
[d,index] = sort(real(c));    % sort on real part
of eigenvalue
if (~( (d(n){0} & (d(n+1)>0) ))
        errmeg('Can't order eigenvalues')
        return
end
chi = v(1:n,index(1:n));      % select vectors
   with negative  eigenvalues
lambda = v((n+1) : (2*n),index (1:n));
s = -real (lambda/chi);
k = r\b'*s;
```

In "lqr", functions "abcdcheck" and "nargcheck" check the dimensions of (A,B,C,D) matrices and check the number of input arguments, respectively. A careful look at the last ten statements of lqr reveals that it is effectively the same as "lq.mtl". Now, suppose that one would wish to write a new .m file to solve the linear state regulator problem, find step responses of the open-loop as well as the optimum closed-loop responses, and plot the outputs. This function, called "Design" is given below:

```
function design (a,b,c,d,q,r,to,dt,tf)
%  Function to design a linear regulator
%  problem and plot the step responses
[f,k] = lqr(a,b,q,r);
ac = a-b*f;
time = [to : dt : tf];
%  open-loop step response
yo = step (a,b,c,d,1,time);
%  optimum closed-loop step response
yc = step (ac,b,c,d,1,time);
Plot (time,yo,time,yc)
end
```

This function is used below for a 3rd order system.

```
>> a = [0 1 0 ; 0 0 1 ; .4 .5 .8]; b = [0 ; 1 ; 1];
>> c = [0 ; 1 ; 1 ]; q = eye (3); r = 1; d = 0;
>> design (a,b,c,d,q,r,0,0.1,10.)
```

One can now print some values of the intermediate

```
matrices:
>> k
     k =
          1.7643          0.9498          0.0828
          0.9498          2.1145         -0.5971
          0.0828         -0.5971          2.5808
>> f
     f =
          1.0326         1.5174          1.9837
>> eig(ac)
     ans =
          -1.9078
          -0.3966 + 0.3965i
          -0.3966 - 0.3965i
>> eig(a)
     ans =
          -0.2876 + 0.45631i
          -0.2876 - 0.45631i
           1.3751
```

Since the open-loop System is unstable, the open-loop response is scaled down by 10^{-12} to plot the two response together the resulting plots are shown in Fig. 20.

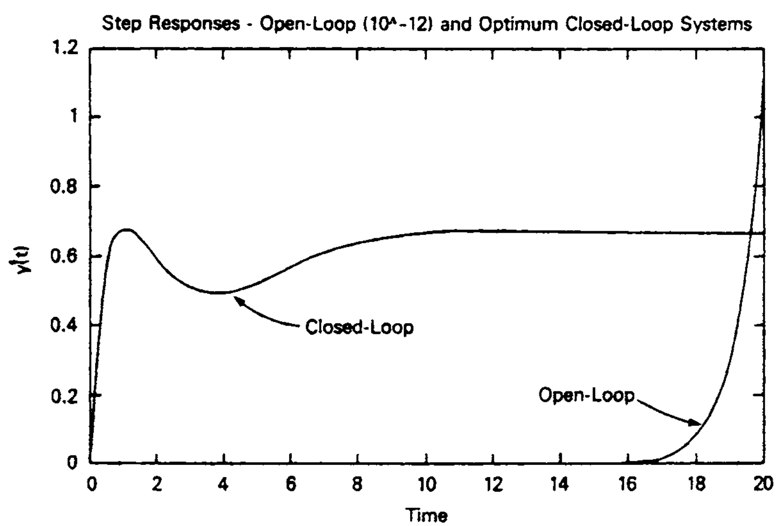


Fig. 20 Output responses for Example 10.

E. MATLAB Toolboxes and Extensions

Ever since PC_Matlab and PRO-Matlab have been in the CACSD market, many satellite packages to this package have appeared. These packages have been created through a sequence of .m files which have been made either with close collaboration with the Mathworks, Inc. known as "tools boxes" or by independent teams at universities and research institutions. In this section, a brief introduction is provided for this group of CACSD packages.

1. **System-ID tool box.**

One of the first significant extensions to MATLAB was the System Identification Tool Box developed by Ljung[39]. This tool box is a collection of .m files which implement the most common and useful parametric and nonparametric system identification methods. The tool box follows closely the theory developed by the author's book [40]. The first edition of the tool box contains eight functions which can create mathematical models based on input/output data. These functions are:

1. ARMAX Auto regression, moving average with extra input approach
2. ARX Autoregressive with extra input
3. BJ Box-Jenkins approach
4. IV Instrumental variable method for an ARX-model structure
5. IV4 Estimation of the parameters of an ARX-model structure using a near-optical four-stageinstrumental variable procedure
6. OE The output-error model identification method
7. PEM Predictor-error model-Estimation of the parameters of a general MISO linear model Structure
8. SPA Spectral Anaylsis of linear discrete-time systems

This tool box has proven to be a very successful addition to the original MATLAB. For example, Jamshidi, et al. [41] have used it for the identification of the models for a very complex three-channel multiple mirror telescope testbed system using experimental data gathering for input and output of the system components. Input/output data files were created experimentally and then loaded into MATLAB environment for component model identification.

2. **Robust control tool box.**

Another useful extension to MATLAB has been the Robust Control Tool Box developed by Chiang and Safonov[42]. The tool box would allow the user to design a "robust" multivariable feedback control system based on the concept of the singular-value Bode plot. The following "tools" are included in this MATLAB extension:

1. LQG-based optimal control synthesis including loop transfer recovery and frequency-weighted LQG.
2. H_2 and H_∞ optimal control synthesis.
3. Singular-value-based model reduction for robust control including balanced truncation, optimal Hankel, and balanced stochastic truncation.
4. Multivariable digital control system design.

The toolbox contains a number of demonstration examples including the elevon and canard actuarot control of a fighter aircraft system and modal control of a large-space structure. This extension of MATLAB has certainly satisfied a need in robust control system design, which is a popular research topic in today's control theory.

The basic commands featured in this toolbox are

H2QG	H_2 optimal control synthesis
HINF and LINF	H_∞ optimal control synthesis
LQG	Linear quadratic Gaussian optimal control synthesis
LTRU and LTRY	LQG loop-transfer recovery optimal control synthesis
YOULA	LQG control synthesis using Youla parametrization

There are perhaps a few more toolboxes and MATLAB extensions which may be missing here. One such toolbox is the "State-space identification tool" by Milne [43] which provides MATLAB-like commands to create continuous-time linear-time-invariant models using the maximum likelihood scheme. In the next three sections, three more toolboxes are discussed very briefly.

3. Large-scale system tool box.

One of the attempts to create design-specific tool boxes is underway at the University of New Mexico CAD Laboratory for Intelligent and Robotic Systems made for analysis, simulation, and design of large-scale systems, based on the book by Jamshidi [44]. In this software environment, various issues of large-scale systems such as model reduction, analysis structural properties, control, and design are incorporated. The basic features of this CACSD environment are similar to LSSPAK/PC , which is discussed in some detail in Section IV. Details regarding this tool box can be obtained from the author.

4. The control kit.

Another good example of the extendability of MATLAB is the "Control Kit", which has been created by the research team at the University of Sussex through the efforts of Atherton, et al. [45]. The main objective of the Control Kit is to create a menu interface for a classical control course to accompany PC-MATLAB for students without the knowledge of this package. Although any menu-driven package is bound to have some inevitable restrictions, the Kit exploits the total flexibility of MATLAB. The theme structure of the control system in the Kit is a SISO system with a plant $G(s)$, controller compensator $G_c(s)$, and a feedback block $H(s)$. Through the various layers of menu, the users can build their own structure, realize it in both frequency and time domains, analyze it and try a number of different controller structures. Clearly, every feature of linear system analysis and design of SISO systems that MATLAB supports is also supported by the Control Kit. The Kit can be obtained by contacting the authors [45]. Atherton's group has also created a MATLAB-based series of .m files for nonlinear 3180 systems analysis including the describing function method.

5. Robotics tool box.

Extension of MATLAB is not restricted jus to control and identification. Once again at CAD Laboratory for Intelligent and Robotic Systems of the University of New Mexico, in collaboration with Tampere University of Technology, Finland, a new effort has started to create an environment for robot manipulators. The environment contains a library of robots such as PUMA 560, Adept II, Rhino XR-2, and so on. It handles all the basic problems for robotics such as kinematics (forward and inverse), dynamics (nonlinear and linear), trajectory planning, simulation, analysis, and control. The initial report on this project can be found in Honey [46] as well as Honey and Jamshidi [47].

IV. OTHER DESIGN PACKAGES

In this section, few CACSD packages that have been developed independently of MATLAB are presented. It should be noted that due to limited space, one cannot cover all such packages.

A. FREDOM - TIMDOM - LSSPAK

The past eight years has been an especially active period for computer-aided design of control systems in the U.S. and Europe. The use of the computer as a design tool in integrated electronic circuits has become very common in both the industry and academic institutions. However, in the area of control system theory the past years has been very ciritical. Today, there is hardly any university which does not have access to CAD software for control systems. Toward this goal, the Computer-Aided Design Laboratory for Intelligent and Robotic Systems (CAD LAB) in the Electrical and

Computer Engineering Department of The University of New Mexico has been active in developing and implementing various CAD software environments for design and analysis of linear control systems, robot manipulator control and simulation. One such CAD software program CONTROL.lab/VAX, was already described briefly. In this section, three other software packages - FREDOM, TIMDOM and LSSPAK on the HP 9800 series as well as on the IBM PC and its compatibles will be briefly described. These packages are

(1.) FREDOM - A CAD package for linear classical control systems
(2.) TIMDOM - A CAD package for linear modern control systems
(3.) LSSPAK - A CAD package for large-scale linear control systems

1. **FREDOM.** FREDOM is a FREquency-DOMain CAD package for SISO (single-input single-output) systems described by a pair of transfer functions described by a feed-forward transfer function G(s), described by [44].

$$G(s) = \frac{Af(s)}{Bf(s)}$$

$$= \frac{s^m + a_{m-1}a^{m-1} + \cdots + a_1 s + a_0}{s^n + b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}$$

and a feedback transfer function,

$$H(s) = \frac{Cf(s)}{Df(s)}$$

$$= \frac{s^p + c_{p-1}s^{p-1} + \ldots + c_1 s + c_0}{s^q + d_{q-1}s^{q-1} + \ldots + d_1 s + d_0}$$

or they may be represented in the z-transform domain for discrete-time systems as shown by:

$$G(z) = \frac{Af(z)}{Bf(z)}$$

$$= \frac{z^m + a_{m-1} z^{m-1} + \cdots + a_1 z + a_0}{z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0}$$

$$H(z) = \frac{Cf(z)}{Df(z)}$$

$$= \frac{z^p + c_{p-1} z^{p-1} + \cdots + c_1 z + c_0}{z^q + d_{q-1} z^{q-1} + \cdots + d_1 z + d_0}$$

where $m \leq n$ and $p \leq q$. There is no loss in generality by assuming,

$$a_m = b_n = d_p = d_q = 1$$

since any of these coefficients can be normalized to one.

In sequel, a brief description of the capabilities of FREDOM will be presented.

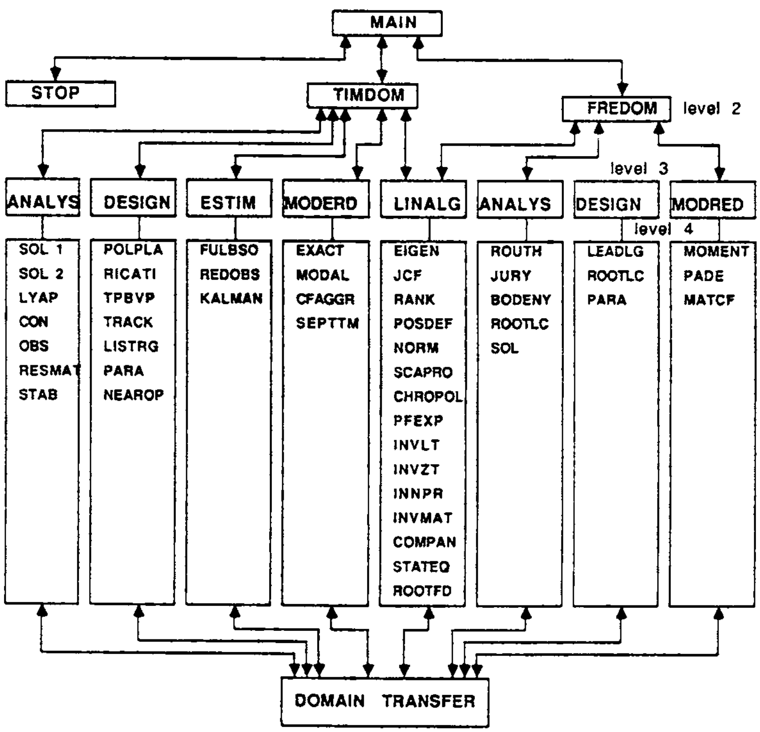


Fig. 21. A tree structure for FREDOM-TIMDOM/45.

a. Analysis: The analysis of linear control systems in FREDOM can be performed with several techniques. Fig. 21 shows a tree structure of FREDOM-TIMDOM/45--the version on the HP9845 computer. The structure of FREDOM and TIMDOM on other computers such as the IBM PC/XT may differ to some extent. However, the essential elements of both versions of the packages are somewhat the same.

When "ANALYS" is selected FREDOM/45 a menu with the following options will apear:

COMMAND	- Description
ROUTH	- Routh-Hurwitz stability criterion
JURY	- Jury-Blanchard stability
BODENY	- Bode-Nyquist plots
ROOTLC	- Root-Locus plot
SOL	- Time domain solution of transfer function
EXIT	- Leave frequency analysis

The first command, "ROUTH" performs a Routh-Hurwitz stability criterion for a SISO system. The second menu item is "JURY", which will construct the Jury-Blanchard table and, if necessary, the Raible table, for discrete-time system stability analysis. The remaining items are similarly self-explanatory.

The final analysis technique is "SOL", which provides the complete response of the system. This is done by converting G(s) and H(s) into a state equation and appropriate integration and plotting routines are used to simulate the system. The above menu items will appear in other version of FREDOM, e.g. FREDOM/PC/XT. However, the name(s) of some commands may be different on the PC version. For example, "SOL" in FREDOM/45 is called "COMSCS" in FREDOM/PC.

b. Design: The design of systems in the frequency domain involve the classical approaches to design of SISO systems. When "DESIGN" in the frequency domain is selected the following menu will appear:

Design was chosen. Your options here are:

COMMAND - Description
LEADLC - Lead, Lag or Lead-Lag Compensation
ROOTLC - Root-locus Plots
PARAOP - Parameter Optimization
EXIT - Leave frequency design

Once again, the above items are self-explanatory.

c. Model Reduction: The methods of model reduction in the frequency domain are restricted to SISO systems except for the matrix continued fraction method. When the user selects "MODRED" in the frequency domain the following menu will appear:

COMMAND - Description
PADE - Pade Approximation
MOMENT - Moment Matching
MATCF - Matrix Continued Fraction
EXIT - Leave frequency domain model reduction

It should be noted that both FREDOM/45 and TIMDOM/45 as depicted in Fig. 21 in extended BASIC source are completely listed in a linear systems book by Jamshidi and Malek-Zavarei [49].

2. **TIMDOM.** TIMDOM is a TIMe-DOMain CAD package for MIMO systems described by a quadruple of matrices (A,B,C,D) representing a linear system in state-space form:

$$\dot{x} = Ax + Bu, \; x(0) = x_0$$

$$y = Cx + Du$$

where A is n x n system matrix, B is n x m input matrix, C is r x n output matrix, D is r x m input-output matrix, and x, u, and y are n x 1 state, m x 1 control and r x 1 output vectors, respectively and x_0 is the initial state. In sequel, a brief description of the capabilities of TIMDOM will be presented.

a. Analysis: The analysis of linear control systems in TIMDOM can also be performed with several techniques. When "ANALYS" command of TIMDOM is provoked the following options appear:

COMMAND	Description
SOL1	Analytical solution of state transition matrix
SOL2	Numerical solution of the state equation
LYAP	Lyapurov equation solution
CON	Controllability check
OBS	Observability check
RESMAT	Resolvent-matrix in Q(s)/P(s) form
STAB	Stability of the origin for continuous time and discrete-time systems
EXIT	Leave time domain analysis

The above commands are fairly self-explanatory.

b. Design: Upon selection of "DESIGN" in the time domain the user is presented with the following menu:

COMMAND	Description
POLPLA	Pole placement
RICATI	Riccati equation solution
TPBVP	Two-point boundary-value problem
TRACK	Tracking problem
LISTRG	Linear state regulator problem
PARA	Parameter optimization
NEAROP	Near-optimum design
EXIT	Leave time domain design

The first item, "POLPLA", is concerned with the placement of poles for a closed-loop system, which provides a state feedback controller for a linear time-invariant SISO system. The remaining 5 commands are self-explanatory. The command "NEAROP" refers to the case where a controller (say state feedback) is designed for a reduced-order (aggregated) model. c. Model Reduction: The model reduction schemes of TIMDOM fall along the lines of large-scale systems order reduction schemes [44] "aggregation" and "perturbation".

When provoking the "MODRED" command, the following sub-commands would typically appear in TIMDOM:

COMMAND	Description
EXACT	Exact aggregation
MODAL	Modal aggregation
CFAGGR	Continued fraction aggregation
SEPTIM	Separation of time scales
EXIT	Leave time domain model reduction

Here, the first three commands are typical time-domain aggregation schemes, while the last one, "SEPTIM" corresponds to a perturbation method, whereby a system is checked whether its variables' TIMe scales can be SEParated into a "slow" and a "fast" variables.

d. Estimation/Filtering: TIMDOM provides two techniques for state estimation and one for filtering. In a typical use of estimation/filtering submenu, called "ESTIM", the following options appear:

COMMAND	Description
FULOBS	Full order observer
REDOBS	Reduced order observer
KALMAN	Kalman filtering
EXIT	Leave time domain estimation/filtering

The first item is essentially the design of a state estimator of the Luenberger-type for a linear time-invariant SISO system. The second one is the reduced-order version of it whereby only (n-r), when n is the number of states and r is the number of outputs, of the system's state variables are estimated. The last item is the design of a Kalman filter with a zero-mean white noise driven linear time-invariant discrete-time system.

3. **LSSPAK** LSSPAK is a CAD package for modeling and control of large-scale linear systems. The name LSSPAK stems from Large-Scale Systems PAcKage. It consists of four main submenus: Linear Algebra (LINALG), Model reduction (MODRED), Analysis (ANALYS), and Design (DESIGN).

a. Linear Algebra/(LINALG) The linear algebra programs supported by LSSPAK is essentially the same as TIMDOM or FREDOM with exceptions of: (i) "Cheby"

--a Chebyschev polynomial curve fitting program and (ii) "Genrank" --a program to calculate the generic rank of a structured matrix for use in checking the structural controllability and observability of a large-scale linear interconnected system.

b. Model Reduction (MODRED) Under model reduction, LSSPAK supports both frequency-domain techniques of FREDOM and time-domain methods of TIMDOM. In addition it provides:

PADMOD - Pade-Modal method
PADROUT - Pade-Routh method
CHAIN - Chained aggregation
BALANC - Balanced realization

c. Analysis (ANALYS) In the analysis sub-menu of LSSPAK, the following options are offered:

COMMAND	Description
STAB	Stability of a large-scale system via the Lyapurove method
STRCON	Structural controllability
STROBS	Structural observability
SIMUL	Simulation of a large-scale system

d. Design (DESIGN) The design of a large-scae systems can take on two basic forms: "hierarchical" control, which is associated with the decomposed form of a large-scale system and "decentralized" control which is associated with the decentralized form. Here, the following options are offered:

COMMAND	-	Description
GOALCR	-	Goal coordination algorithm of hierarchical control
INTPRD	-	Interaction prediction algorithm of hierarchical control
DYNCOM	-	Dynamic Compensation of decentralized control
ROBDEC	-	Robust decentralized controller design
DECSTM	-	Decentralized stabilization via the multi-level method

Throughout this chapter, several examples on the use of TIMDOM/PC and LSSPAK/PC have been presented. Below, we conclude this section by presenting one more example.

Example 11: In this example, a third order system with two inputs and one output issimulated on TIMDOM/PC.

```
<<COMSCS>> Finds a COMplete Solution of a linear
Continuous-time  System  via
4th order Runge-Kutta with graphical plots
ORDER of the system n = 3 No. of system INPUTS m = 2
No. of system OUTPUTS r = 1
Initial time to = 0 Final time tf = 5 Step size dt = .1

Matrix A
     0.000E+00  0.100E+01  0.100E+01
     -.200E+01  -.200E+01  0.100E+01
     0.000E+00  0.000E+00  -.200E+01
Matrix B
     0.100E+01  0.100E+01
     0.000E+00  0.100E+01
     0.100E+01  0.000E+00
Matrix C
     0.100E+01  0.500E+00  0.250E+00
Matrix D
     0.000E+00  0.000E+00
Initial STATES:
 1   0.000E+00
 2   0.000E+00
 3   0.000E+00
```

The output responses of the system for inputs $u = (1\ 0)^T$ and $u = (0\ 1)^T$ are shown in Fig. 22.

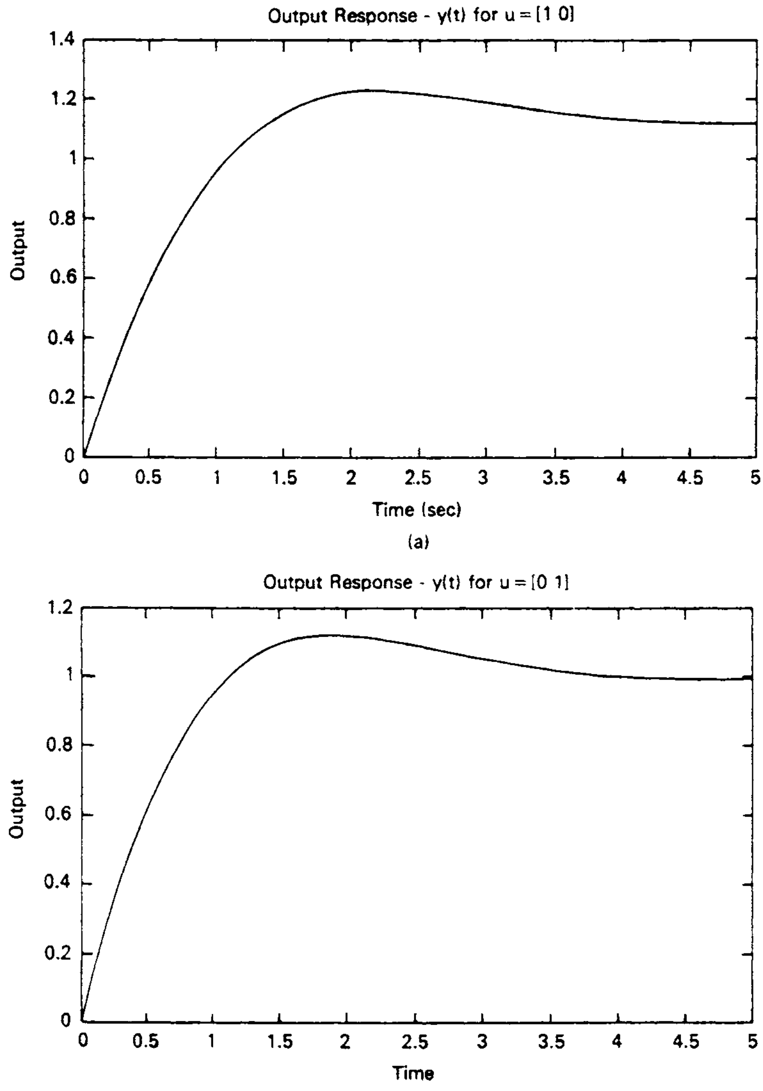


Fig. 22 Output response of Example 11
(a) u = [1 0] and (b) u = [0 1]

B. CADACS (KEDDC)

CADACS is comprehensive CAD package designed to cover a wide range of control system engineering tasks. It contains modules for process identification, system analysis, controller design, simulation and controller implementation based on a broad variety of modern as well as classical approaches for SISO and MIMO systems. This package is regarded as having a user-friendly interface, graphic output and state-of-the-art numerical algorithms. This package is based on the KEDDC framework, developed at Ruhr-University Bochum, Germany [23]. CADACS is available in different configurations and versions. It can easily be tailored and adapted by the user, as it is written in FORTRAN. CADACS can be used in the form of complete main programs or in the form of a comprehensive control engineering library. It provides a system which minimizes engineering and programming resources required for the complete cycle of system identification, control design and design verification. Table X shows the features of CADACS. The various systems' descriptions, forms, and transformations supported by CADACS are shown in Fig. 23. Various classes of techiques supported by CADACS are breifly described.

Identification Methods

- Approximation in time domain (6 methods)
- Approximation in frequency domain (2 methods)
- Approximation by exponential functions
- Approximation by momentum methods
- Recursive least squares algorithms
- Correlation and spectral analysis
- Maximum-likelihood parameter estimation
- Various parameter estimation methods using different numerical methods.

Control System Design Methods

- Continuous and discrete time compensators
- Finite settling time methods
- Various pseudo-compensator methods
- Optimization of PID-type controllers
- State-feedback controllers with or without PI-action
- Inverse-Nyquist-array technique
- Reduced and full-order observers
- Model reduction in open and closed-loop configurations

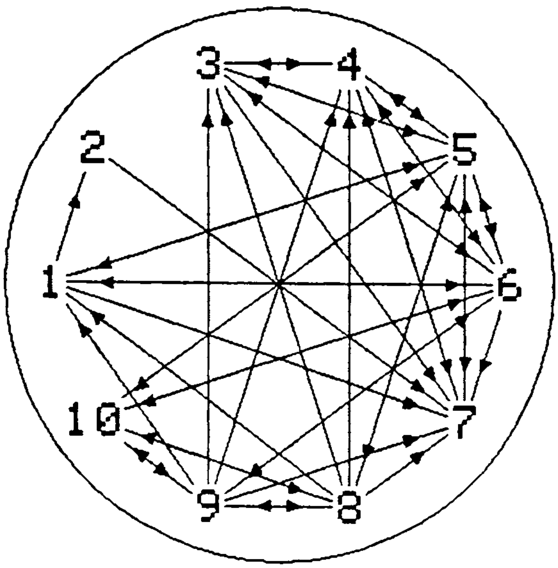
- Parameter optimization using sets of performance indices and design criteria
- Kalman filter design

Adaptive Control System Design Methods

- Adaptive PI-controller using periodic test signals
- Adaptive compensator using various parameter estimation methods
- Self-tuners
- Various model reference adaptive control approaches
- Adaptive observers
- Multivariable adaptive controller design
- Adaptive controller simulation

TABLE X. Features of CADACS (KEDDC).

- Simple command-driven dialog combined with questions and answers
- Hierarchy of independent program modules
- Control engineering library
- Reliable numerical algorithms drawn from recent research in numerical analysis
- Device independent graphics output.
- Transformation of system representation forms using alternative sets of strategies or numerical algorithms
- Signal analysis and filtering
- Off-line and on-line identification using deterministic and stochastic approach (20 methods)
- Model reduction
- Controller design (15 approaches)
- Parameter optimization
- Observer design (pole placement, Kalman filter)
- Simulation (general and special purpose simulators)
- Adaptive controller design (5 approaches)
- Controller implementation (simulation, real time)
- Standard tasks centralized in 'Manager' programs



1. series of deterministic or stochastic input and/or output signals,
2. series of auto- or crosscorrelation values,
3. discrete values of the pulse response,
4. discrete values of the step response,
5. transfer function (matrix) in s-domain,
6. transfer function (matrix) in z-domain,
7. discrete values of frequency responses,
8. matrices for continuous state space,
9. matrices for discrete state space,
10. polynomial matrices.

Fig. 23. Various systems' configuration supported by CADACS (KEDDC).

1. Simulation. Standard structures for linear control systems are simulated by a special simulator. The overall system to be simulated may be composed of different subsystems of different representation forms. Each block in this structure may be form 5,6,8,9 or 10. A second type of simulator performs block oriented continuous-time simulation using elementary CSMP-like blocks. For

direct digital control operation (DDC) different special simulators are available. They simulate the DDC environment for testing DDC algorithms in real-time applications. In all cases graphic output is controlled by a Graphics Manager prepared menu.

2. Centralized Tasks. Frequency used tasks, like transformations to different representation forms, data base definitions or basic calculations, are separated from the other level-3 programs and are concentrated in main interactive system handlers, the so-called Managers:

- Signal Manager for handling of signal sequences
- System Manager for handling of systems described by transfer functions or transfer matrices in s- or z-domain
- Frequency Manager for handling of frequency responses or spectra in tabular form
- Matrix Manager for handling of coefficient matrices or systems in state space
- Polynomial Matrix Manager for handling of polynomial matrices and related system representations
- Graphics Manager for all graphics operation.
- Documentation Manager for real-time signal presentation
- Monitor for central organization problems like task scheduling

CADACS program structure is shown in Fig. 24. As seen, it is organized in a four-level hierarchy of independent program modules. All basic functions are contained within the first level. These include all numerical algorithms (e.g. for matrix or polynomial operations). The second level represents the control engineering library containing all relevant specialized algorithms as subroutines which make extensive use of modules of proven numerical software. Level-1 and level-2 subroutines together form the complete library system. This is one form in which CADACS can be used. Modules of the third level are designed for dedicated main tasks, in which a set or a class of methods is grouped together. They are complete interactive or real-time programs which can run as stand-alone programs or which can run supervised by a central monitor which resides in the top level. It manages the housekeeping services for all internal resources and forms the friendly user interface. Any choice of level-3 programs is possible, but with a proper choice from a large set of programs (some are prerequisite) a CADACS system can be tailored according to the user's requests. Communication between interactive programs is performed by a data base with standard data formats.

CADACS provides a high degree of portability, which results from the interface of CADACS to the operating system and to hardware-dependent functions. The strict separation of interactive and application code makes this possible. For the implementation in an ohter computer environment only these level-1 modules have to be modified. That level contains a number of device dependent functions supporting the freiendly user interface. For a minimal configuration, dummy routines are available to replace non-essential functions.

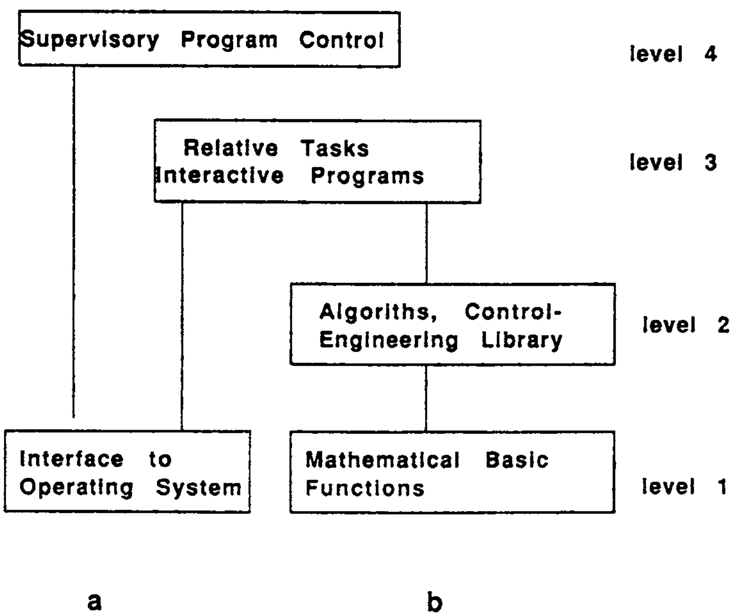


Fig. 24. Program structure of CADACS (KEDDC).

3. Interactive Operation. An unified command-driven dialog combined with question-and-answers allows a high degree of user interaction. Each level-3 interactive program has its own set of commands, which can be enabled by supervisory scheduling of the programs using the monitor. An active program prompts with its name. A two-character command initiates a subtask or a local question-and-answer dialog, where values can be entered in free format. Commands are natural, simple and powerful. The local question-and-answer dialog consists of questions sent by the computer, which ends with ' = ? ', where the user has to specify one of some values by typing an alphanumeric character string. If the question ends with ' ? ' only a yes/no answer is requested. Reasonable defaults make this dialog easy to use. Mistyping is detected and the question is repeated. On blank input default values are assumed. The menu of commands or status informations can be displayed using standard commands. At any stage the user may get 'HELP' information. This is essential for error recovery. The user is provided with a detailed analysis of his errors and an explanation about what happened. In addition he will get some hints, are given on how to proceed. The command interpreter handles both local commands and central commands which central commands which can also be tailored by the user. Level-3 programs are interruptible at any stage, and new subtasks from other level-3 programs can be interlaced into the running dialog if program cloning is supported by the operating system. This feature allows a very free and flexible use of CADACS.

4. Graphics CADACS programs themselves do not generate graphics code. Graphics operations are performed by a central Graphics Manager, which itself is an autonomous package. Data are send from the application programs to the Graphics Manager via level-1 communication interface. The Graphics Manager is a very comfortable special task for formatting data into graphic form readable by the control engineer. The Graphics Manager can be implemented using a graphics processor, or on an intelligent graphics terminal or as an internal

graphics task on the same computer. This concept slows a great deal of flexibility in interfacing to different graphics device configurations and integrates diverse parts of CADACS into an unified system. A task-based Graphics Manager is available in CADACS. It incorporates state-of-the-art device independent graphics which is based on different standards. All diagrams relevant to control engineering are implemented. The user has the choice of using default diagrams or tailor his plots using a keyword-based descriptive language.

The following example illustrates the use of CADACS (KEDDC):

<u>Example</u> <u>12</u>: To illustrate the user dialogue of the package, a hardcopy of some sections of a longer KEDDC session has been copied into this chapter. The following is showing the original dialoque. Operator input is underlined, command lines are printed in bold letters and command menus are framed. Graphics output is not exactly placed in chronological order with the dialogue. The hardcopy is printed when the image is complete.

In order to understand this demonstration, section numbers are added to the left margin. Section (1) starts with Frequency Management. The command menu is shown which appears in a special window. Principal gain plot is requested by rejected as explained using 'help' at (2). The principal gains for a system read from file TGM are plotted at (3). At (4) Matrix Management is interlaced to generate a 3D-mesh surface for the system given in state space description. Simulation is interlaced at (6) and then a second Matrix Management at (7) to display the pole zero pattern of the discretetime system to be controlled. At (8) Simulation is continued. After some configuration dialogue, which is not shown here, the screen shows the results of all sections at (9). The graphic viewports are numbered from left to right beginning at the bottom of the screen. The viewpoint number 5 is a small one which is inside of number 3.

```
(1) KEDDC
    KED66:    FR
    FRM66:    ??

*******************Frequency_Management*******************

    SYSTEM:                                    FURTHER CALCULATIONS
RE  Read system from file                PO   Popov-frequency response
    G(s),G(z),u(t),y(t),G(jw)            IV   Inverse frequency response
    g(t),h(t),(A,B,C)                    WC   Magnitude/phase for spec. freq.
LI  List system datas  1 WH Write step   RA   Magnitude or phase margins
    FREQUENCY_TABLE: response on file    TT   additional deadtime
GO  Generate frequency table             FH   Step response from freq. resp.
EO  Edit frequency table                      CONNECTIONS:
RO  Read frequency table from file       S2   Read 2nd system from file
IO  Input frequency table using dialog   **   Parallel connection
LO  List frequency table                 **   Series connection
WO  Write frequency table on file        //   Negative feedback connection
    FREQUENCY_RESPONSE:                       GRAPHICS:
FG  Calculate frequency response              BO   Bode plot complete
RF  Read frequency response from file         BM   Bode plot magnitude
IF  Input frequency response using dialog     BP   Bode plot phase
EF  Edit frequency response                   NY   Nyquist plot
LF  List frequency response                   NI   Nichols plot
WF  Write frequency response on file          DH   Step response
FM  Create frequency response matrix          TR   Toggle curve tracking on/off
    on file                                   MP   Change no. of dots per dia.
ST  Status protocol                           PG   Principal gain plot
Standardfilenames see next page
    FRM66:    PG
    /FR-004 THERE IS NO FREQUENCY TABLE !
(2) FRM66:    HE
    /FR-004 THERE IS NO FREQUENCY TABLE !
   You have initiated an operation which requires a
   frequency table.  This table isn't defined.
   Read table from file with RO, or specify table
   interactively with IO, or generate it automatically
   with GO.  The last one is the most comfortable way
   if the frequency range is initially unknown.
    FRM66:    GO
    RECOMMENDED FREQUENCY RANGE FROM    1.79416    TO    899.2309
    FREQUENCY RANGE FROM, TO       -  ?  1,1E3
    NUMBER OF FREQUENCY VALUES     -  ?  100
    FRM66:    PG
    FREQUENCYFACTOR  -  ?
    FILENAME         -  ?  TGM::DA
---------------------------------------------------------------------
```

```
    file=TGM       ::DA,  type=UE
    created using SMGR  by SCHMID on  4.  9.1983 at  11.35
    TURBOGEN.  MODEL 1 MIMO
    IT IS A TRANSFER MATRIX
---------------------------------------------------------------------
    NUMBER OF INPUTS  -  2.  NUMBER OF OUTPUTS  -  2
(3) MAXIMUM PRINCIPAL GAIN IS MAPPED TO VIEWPOINT  4, CURVE  1
    MINIMUM PRINCIPAL GAIN IS MAPPED TO VIEWPOINT  4, CURVE  2
(4) FRM66:    @@
    KEDDC:    MM
    MMG66:    RS
    FILENAME - ? VFELDE:_:DA
    file=VFELDE: :DA,  type=MA
    created using MMGR by SCHMID on 31.10.1980 at 13.25
    TURBOGEN. STATE SPACE MODEL A2Y3
      2 BLOCKS,   3 MATRICES IN 1ST BLOCK
      BLOCKNO., MATRIXNO. - ? 2
    MMG66:    MS
    DO YOU WANT GRAPHICS ? Y
    MATRIX A
    + 0 0 0
    0 + 0 0
    0 + + 0
    + 0 0 +
    MATRIX B
    + 0
    0 +
    + +
    + +
    MATRIX C
    0 0 1 0
    0 0 0 1
(5) MATRICES MESH SURFACE IS MAPPED TO VIEWPORT 2, CURVE 1
    MMG66:    EX
(6) FRM66:    @@
    KEDDC:    DG
    DIG66:    @@
(7) KEDDC:    MM
    MMG66:    RS
    FILENAME - ? TZM:_:DA
    file-TZM        : :DA,  type-MA
    created using MMGR  by SCHMID on  9.  9.1983 at 11.46
    TURBOGEN. STATE SPACE MODEL A2YA
      3 BLOCKS,   3 MATRICES IN 1ST BLOCK
      BLOCKNO., MATRIXNO.  -  ?  2
    MMG66:    DM
```

```
     SAMPLING INTERVAL =  ?  .2
     MMG66:    EI
     DO YOU WANT GRAPHICS  2 Y
     ----------------EIGENVALUES----------------
     NO   REALPART          IMAGINARYPART        S/I
     1     .385317            .285704             -
     2     .385317           -.285704             -
     3     .462243            .308177             -
     4     .462243           -.308177             -
     5     .644391            .250971             -
     6     .644391           -.250971             -
     7    2.131015E-02        .437934             -
     8    2.131015E-02       -.437934             -
     9     .715286           0.000000             -
     EIGENVALUES ARE MAPPED TO VIEWPORT  5. CURVE  1
     MMG66.    NU
     DO YOU WANT GRAPHICS ? Y
     --------------------ZEROS--------------------
     NO   REALPART          IMAGINARYPART        S/I
     1    -2.03490            0.00000             +
     2    -1.73257            0.00000             +
     3     .770414            0.00000             -
     4     .368101            .463010             -
     5     .368101           -.463010             -
     6     .449659            .360620             -
     7     .449659           -.360620             -
     ZEROS ARE MAPPED TO VIEWPORT  5.  CURVE  2
     MMG66:    EX
(8)  DIG66:    RE
     FILENAME = ? TGM:_:DA
           .
           .
           .
     DIG66:    SI
     SIMULATION STEP SIZE = ? .1
     SIMULATION TIME        = ? 10
     SIGNAL U ( 1) IS MAPPED TO VIEWPORT  1, CURVE 1
     SIGNAL U ( 2) IS MAPPED TO VIEWPORT  1, CURVE 2
     SIGNAL Y ( 1) IS MAPPED TO VIEWPORT  3, CURVE 1
(9)  SIGNAL Y ( 2) IS MAPPED TO VIEWPORT  3, CURVE 2
```

Figure 25 shows some graphical outputs of CADACS.

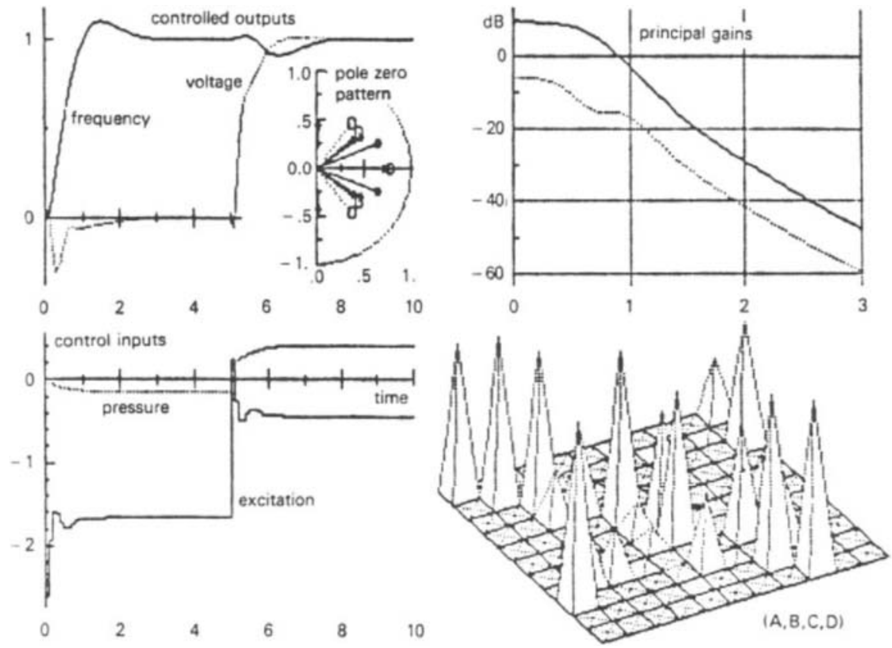


Fig. 25 Some graphical outputs of CADACS software environment.

c. L-A-S

L-A-S (Linear Algebra and Systems), developed by Bingulac and associates [49], is a CACSD package which fully supports a full-fledged programming language to construct, test and evaluate various algorithms for analysis and design of control systems. The fundamental concept behind the L-A-S operator is that the L-A-S operator performs some described calculation on input data, indicates errors, if any, and generates output data (results). The output data of operator may be used as input data by any operator to be executed later.

The operators together with the names of input and output data are issued to the L-A-S language *interpreter* in the form of an L-A-S *operator* *statement*. The sequence of L-A-S operator statements performing a desired analysis/design task is called a *L-A-S* program. A typical example of an L-A-S program implementing the LQG/LTR (linear quadratic Guassian/loop transfer recovery) design methodology is given later on. The following discussions are based on a paper by Bingulac [49].

Due to limited space, only a few statements from the L-A-S program will be explained. It is hoped, however, that the basic feature of the L-A-S language will be grasped and its applicability in solving a wide range of system/control analysis and design problems will become apparent. Particular attention will be paid to the LQG part of the design methodology, wihle the LTR part will only be mentioned briefly. Cited references provide more details.

1. L-A-S language overview. In the L-A-S language there are two types of operator statements:
   1. Single operator statement (SOS)
   2. Multiple operator statement (MOS)

   The syntax of the SOS is as follows:

   <label>:<inp-fl>(<oper0fl>)=<out-fl>

The symbols ":" "(" and ")=" act as field delimiters. The <label>, which is optional, is usually used for recursive calculations. The input field, <inp-fl>, consists of zero or more variables names and/or constants, separated by commas. The operator field, <oper-fl>, contains the operator mnemonic name. The output field, <out-fl>, consists of zero or more variable names defined by the operator. The syntax of the MOS is basically the same as that of the SOS, with the exception of the input field where, in addition to variable names or constants, also the *generalized variables* may be specified.

The syntax for the generalized variable is as follows:

<inp-fl>(<oper-fl>)

Since in operator statements the operator name follows all input variable names, it may be concluded that the L-A-S operator statements are written in the *postfix* notation.

As a simple example of both SOS and MOS, consider the calculation of the matrix S given by: $S = B\,R^{-1}B^{T}$. The SOS approach to the calculation of the matrix S is:

R(-1) = T1	Matrix inversion;	$T_1=R^{-1}$
B(t) = T2	Matrix transportation;	$T_2=B^T$
B,T1(*)=T3	Matrix multiplication;	$T_3=B * T_1$
T3,T2(*)=S	Matrix multiplication;	$S = T_3 * T_2$

The MOS approach allows that all four operators be specified within a single operator statement, that is,

B,R(-1),B(t)(*)(*)=S

The generalized variables in this case are: R(-1); B(t) as well as: R(1), B(t)(*). The rather elaborate example of the MOS, given at the end of the Program 1 later on, deserves special attention. Consider a calculation of the matrix *F* defined by:

$$F = \begin{vmatrix} C^T\ (CC^T)^{-1} \\ -(C\ A^{-1}B)^{-1} \end{vmatrix}$$

Given matrices A, B, C, the matrix *F* may be calculated by the following MOS:

```
C(t),C,C(t)(*)(-1)(*),
        C,A(-1),B(*)(-1),-1(s*)(rti) = F,
```

where the L-A-S operators s^* (scalar multiply) and RTI (Row tie) perform multiplication of a matrix with a scalar and tying two matrices by rows, respectively.

Below is a CAD example presented to investigate the use of L-A-S in inputing two frequency-domain blocks, convert them to state space, cascade them together, and simulate it under an input step excitation.

<u>Example</u> <u>13</u>: In this example, the system of Example 1 shown in Fig. 5 is reconsidered. As seen, the

system consists of three poles and a zero. A listing of an L-A-S session to simulate this system for a step input follows.

```
*  (inp-f; input the coeff. of the first system's
denominator 1,0.4,(dma,t) = f; input coeff. of the
first system's denominator
        f
1.00      .400      1.000
*  (pmi) = g; coeff. of numerator must be in a
polynomial matrix form

Enter dimens. and order for polyn. matrix <g > :
1,1,1 PMF Matrix <g > has < 2> row and < 1> colums
Matrix<g >; Enter E,R,C,D,M,I,Z,P,N or H for HELP :
M
       g
2
1
*    f,g(tfss,t) = a,b,c; convert transfer function
     to state space
Result from the operator <TFSS>;variable <a >

 -.400    1.000
-1.000     .000

Result from the operator <TFSS>;variables <c >
1.000     -2.000

*  1.96,1(dma,t) = f1; input the coeff. of the second
system's denominator
            f1
1.960     1.000

*  (pmi) = g1;coeff. of numerator must be in polynomial
matrix form

Enter dimens. and order for polyn. matrix <g1 > : 1,1,0
PMF Matrix <g1 > has < 1> rows and < 1> columns
            g1
1

*    f1,g1(tfss,t) = a1,b1,c1; convert to state space
Result from the operator <TFSS> ; variable <a1 >

-1.960
```

```
Result from the operator <TFSS> ; variable <b1 >

-1.000

Result from the operator <TFSS> ; variable <c1 >

-1.00

*  a,b,c,a1,b1,c1(ccon,t) = a2,b2,c2; cascade the 2
systems together

Result from the operator <CCON> ; variable <a2 >
-.400     -.706     -.707
1.414      .202     1.980
 .000     -.020     -1.980

Result from the operator <CCON> ; variable <b2 >

     1.000
      .000
      .000

Result from the operator <CCON> ; variable <c2 >

     .000      .707      -.707

*  a2,b2,c2(mtf,t) = f2,g2; convert the result back to
a transfer function

Characteristic Polynomial <f2 >
          f2
     1.960     1.784     2.360     1.000

Transfer Function Matrix <g2 >
          g2
     2.000
     1.000
      .000

Polynomial Matrix <g2 > has   1  Columns

*    20(step,sub) = u; input a unit step response

*  (dsc) = t; input a scalar for the total time of
   simulation
          t
```

10

```
*  f2,g2,u,t(rct) = y; compute the response y of the
   system

*    ylab,_SYSTEM_TIME_RESPONSE

*    xlab,_TIME

*    y(dis) =
```

Fig. 26 shows a printer plot of the output response of the system. Fig. 27 shows the plot of the simulated system's output. Note the closeness of this response with the dashed plot of Figure 6 obtained by simulating the same system by CTRL_C.

```
*   y(plt) = ; plot the response of the system
Column #    1 of the variable <y      >
                           min.value                                    max.value
                              -.000                                         3.193
1          .000 I*                                                                                                      I
2          .352 I---------------------*                                                                                 I
3          .796 I------------------------------------------------*                                                      I
4         1.205 I-------------------------------------------------------------------------------*                       I
5         1.487 I-----------------------------------------------------------------------------------------*             I
6         1.596 I-----------------------------------------------------------------------------------------------------*I
7         1.540 I------------------------------------------------------------------------------------------------*      I
8         1.362 I------------------------------------------------------------------------------------*                  I
9         1.130 I--------------------------------------------------------------------------*
10         .910 I----------------------------------------------------------------*
11         .754 I-----------------------------------------------*
12         .687 I------------------------------------------*
13         .759 I-------------------------------------------*
14         .798 I----------------------------------------------------*
15         .919 I----------------------------------------------------------------*
16        1.037 I---------------------------------------------------------------------*
17        1.124 I-------------------------------------------------------------------------------*
18        1.163 I---------------------------------------------------------------------------------*
19        1.156 I---------------------------------------------------------------------------------*
20        1.112 I-----------------------------------------------------------------------------*
```

Fig. 26. Output response of system of Example 13 via printer plot using L-A-S.

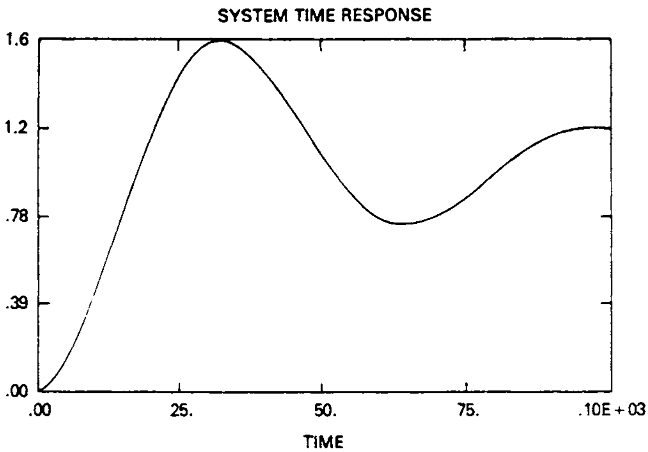


Fig. 27. Output response of Example 14 system via screen plot using L-A-S.

Example 14: In this example, the third-order system described at the end of Example 8 concerns the numerical inversion of the Laplace transform. An implementation of this problem on the L-A-S follows:

```
*  (inp) = f ; input the coefficients of the
   denominator

Enter dimensions for <f   > : 1,4
Matrix <f   >; Enter E,R,C,D,M,I,Z,P,N or H for HELP : m
         f
5,5.25,5,1

*  (pmi) = g ; coeff. of numerator must be in a
polynomial matrix form

Enter dimens. and order for polyn. matrix <g  > : 1,1,3
Matrix <g   >; Enter E,R,C,D,M,I,Z,P,N or H for HELP : m
         g
0
1
1
*    (inp) = t0 ; input the initial value vector

Enter dimensions for <t0   > : 3,1
Matrix <t0    >; Enter E,R,C,D,M,I,Z,P,N or for HELP : m
         t0
1
-4
14.75
```

```
* (dsc) = t ; input a scalar for the total time of
          t
  simulation

10

*  f,g,u,t,t0(rct) = y ; compute response Y of the
   system
```

A printer plot of the numerically inverted output of the system is shown in Fig. 28a. Once again, three simple L-A-S commands will give us a display plot of the response which is shown in Fig. 28b.

There are many other non-MATLAB packages which one could also try to describe. The most notable ones are program CC by Thompson and Wolf [26], package LUND [50]. However, due to lack of space, we only give a brief description of each.

Program CC [26] is an IBM-PC type CACSD package which has been implemented in executed BASIC. It can treat multivariable systems described in both time and frequency-domain, including "transfer function matrix" form. It should be noted that among MATLAB-based program only IMPACT [31] and program M by Lawrence Livermore National Laboratory [51] have this feature. Special features in CC are graphical display of transfer functions, partial fraction expansion, inverted Laplace and z-transform functions, symbolic manipulation of transfer functions, and state space and frequency domain analysis of multirate sampled-data systems. The symbolic manipulation of transfer functions is a rather unique feature of CC. A typical example of this manipulation follows.

```
CC>enter,1,2,5,.5,5,2,1,1,2,2,1,2,5
```

$$G(S) = \frac{5s + .5s + 5}{(s + 2) (s^2 + 2S + 5)}$$

```
CC>pfe
           4.8           .2S - 9.5
G(S) = ─────────  +  ──────────────────
          s + 2       [(s+ 1) +  2 ]

CC>ilt
     CAUSAL INVERSE LAPLACE TRANSFORM
         ⎧ 4.8*exp(-2t) - 4.854122*sin( 2t - 4.121375E-02)
G(t) =  ⎨                               *exp(-1t) for t>0
         ⎩ 0 for t<0
```

Program CC, in its most receng version, represents a very flexible, user-friendly, and powerful package. It deserves a separate section to describe it fully.

V. CACSD TOOLS - A BRIEF SURVEY

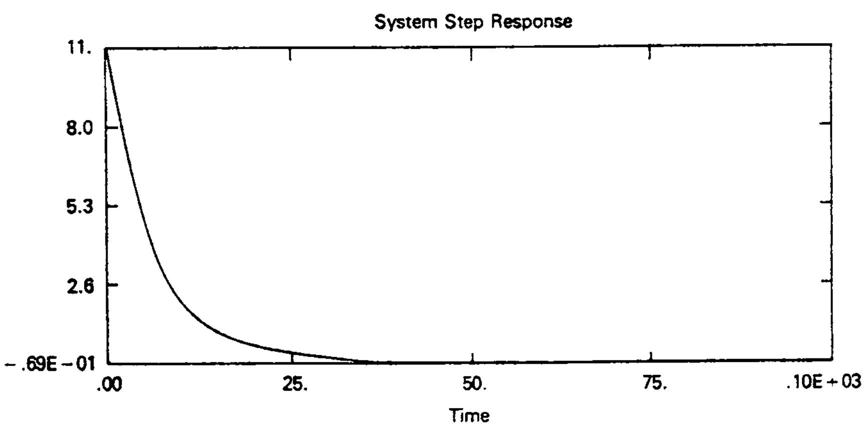
In this chapter, an attempt has been made to give a brief introduction into an increasingly vast field of computer-aided control systems design. It is estimated that over 45 different packages f different degrees of development exist in the world today. A true survey of all these packages is, in fact, a near impossibility. Cellier and Rimvall [2] have made an extensive effort to survey as many CACSD as they could gather. In this section, we present a brief survey of some 22 CACSD packages and try to give a perspective on their usefulness.

Table XI. shows a brief survey on 22 CACSD packages. This table is an abbreviated version of a more extensive one of cellier and Rimvall [2]. We have chosen twelve aspects of CACSD packages - the first six are considered as basic attributes which should be available in all packages. The next five attributes are considered as more advanced ones which may not be available in all programs. The final attributes is a matter of choice on the part of the developers of the particular program.

TABLE XI. A brief survey of 22 CACSD Packages.

```
                    *    ylab, _ System _ Step _ Response
                    *    ylab, _ Time
                    *    y(dis) =

*    y(plt) = ; plot the response of the system

Column #     1 of the variable <y      >

             min.value                                        max.value

                  -.069                                          10.750

1   10.750   |-----------------------------------------------------------*|
2    4.294   |----------------------------*                               |
3    1.840   |------------*                                               |
4     .839   |-------*                                                    |
5     .376   |--*                                                         |
6     .134   |-*                                                          |
7     .004   |*                                                           |
8    -.055   |*                                                           |
9    -.069   |*                                                           |
10   -.057   |*                                                           |
11   -.034   |*                                                           |
12   -.011   |*                                                           |
13    .006   |*                                                           |
14    .014   |*                                                           |
15    .015   |*                                                           |
16    .012   |*                                                           |
17    .007   |*                                                           |
18    .002   |*                                                           |
19   -.001   |*                                                           |
20   -.003   |*                                                           |
```

(a)



(b)

Fig. 28. L-A-S plots for CAD Example 16.

(a) Line printer.
(b) Graphic output plot.

Among the MATLAB-based programs, this particular survey reveals that MATRIXx is the most complete CACSD package. However, three packages CTRL-C, PC_(PRO_) MATLAB, and IMPACT are close behind as second best program. Then comes CONTROL.lab as third best, and fifth overall. Note that the original MATLAB received a mere four points which reinforces the fact that MATLAB was not intended for control systems design.

Among the non-MATLAB packages the results are not surprising. Comprehensive packages such as CADACS (KEDDC) and LUND scored very high indeed. Most other packages are somewhat equal in strength. It is noted that among all packages in Table XI, CADACS of Schmid [23] ranks as the top CACSD program available today. Interested users outside Germany and Austria, may contact the author for the availability of CADACS and L-A-S.

The survey presented here is by no means an objective one. This survey was in fact a rather subjective one. A truely objective survey for a field such as CACSD may not be possible since all such programs and packages are not available to a reviewer and they are often changing in characteristic and capabilities. Moreover, the use of 0-1-2 points system used in Table XI is somewhat arbitrary and a different point system could potentially draw different conclusions. This survey, as indicated earlier, is by and large, based on the responses to a questionaire conducted by Cellier and Rimval [2].

8.6 FUTURE OF CACSD PROGRAMS

The computer-aided design of control system has come a very long way in a few short years. In fact it was only seven years ago when the first comprehensive survey on the subject was published by Jamshidi and Herget [1] which was subsequently translated into Russian. A new edition of this book is under preparation [32]. However, thus far the emphasis on CACSD packages has been on the program, and not so much on the data. The next generation packages should make use of clear distinction between the program (static codes in memory that changes its content during program execution). The future packages should incorporate the user interface as a very important element in the success of their wide-spread use.

In recent years, the integration of control theory and artificial intelligence in various forms such as "connection networks", "neural networks", "expert systems", or "rule-based systems" have become a very popular topic, called by some, "intelligent control". In a similar fashion, CACSD packages are now being developed with the help of artificial intelligence. One such effort is by Robinson and Kisher [28] who have developed an object-oriented AI-based CAD package for nuclear reactor control. They have established it in LISP (LISt Programming) on a LISP machine TI Explorer, and on a regular MACII and MAC/SE. With the use of LISP, CACSD packages would be capable of modifying their codes automatically. However, the drawback is the slower speed of LISP and its difficult user interface.

Another potentially big impact on CACSD environment would come about when parallel computer architecture becomes reality. This would open way to many other possibilities such as the use of PROLOG in its more efficient characteristic. Finally, non-numerical controller designs are perhaps becoming a reality in the future. The emergence of new symbolic languages such as MACSYMA [52], REDUCE [53] and MuMath [54] would help towards that goal.

BIBLIOGRAPHY

1. M. Jamshidi and C.J. Herget, *Computer-Aided Control Systems Engineering,* (eds) Amsterdam: North Holland (1985).
2. F.E. Cellier and C.M. Rimvall, "Computer-aided Control System Design: Techniques and Tools," in *Systems Modeling and Computer Simulation*, N.A. Kheir, ed., Appendix A, New York: Marcel and Dekker (1988).
3. G.A. Korn and J.V. Wait, *Digital Continuous System Simulation*, Englewood Cliffs, NJ: Prentice Hall, 212 p. (1978).
4. IBM, *"Dynamic Simulation Languages/VS (DSL/VS)*. Language Reference Manual," Program Number 5798-PXJ, Form SH20-6288-0, IBM Corporation, Dept. G12/Bldg. 141, 5600 Cottle Road, San Jose, CA 95193 (1984).
5. M.S. Wai and M.H. Dost, *DSL/DP Digital Simulation Language-Durable Precision*, IBM Corp. Gen. Prod. Div., San Jose, CA (1980).
6. E.E.L. Mitchell and J.S. Gauthier, *ACSL: Advanced Continuous Simulation Language-User/Guide Reference Manual*, Concord, MA: Mitchell & Gauthier, Assoc. (1986).
7. D.C. Augustin, M.S. Fineberg, B.B. Johnson, R.N.Linebarger, F.J. Sanson, and J.C. Strauss, "The SCI Continuous System Simulation Language (CSSL)," *Simulation*, 9, 281-303 (1967).
8. C. Moler, *MATLAB User's Guide*, Department of Computer Science, University of New Mexico, Albuquerque, NM (1980).
9. M. Jamshidi, "An Overview on the Algebraic Matrix Riccati Equation and Related Problems," *J. Large-Scale Systems,* vol. 1, pp. 167-172 (1980).
10. J.J. D'Azzo and C.H. Houpis, *Feedback Control System Analysis and Design,* New York: McGraw Hill (1966).
11. H.H. Rosenbrock, "Design of Multivariable Control Systems Using the Inverse Nyquist Array," *Proc. IEEE*, vol. 116, pp. 1929-1936 (1969).
12. W.A. Wolovich, *Linear Multivariable Systems*, New York: Springer Verlag (1974).
13. J. Ackerman, *Sampled-Data Control Systems*, Berlin: Springer Verlag (1985).

14. U.Y.D. Landau, *Adaptive Control: The Model Reference Approach*, New York: Marcel Dekker (1979).
15. V.V. Chalam, *Adaptive Control Systems*, New York: Marcel and Decker (1987).
16. P. Dorato, *Robust Control*, New York: IEEE Press (1987).
17. R. Dorato and R.K. Yedavalli, *Recent Advances in Robust Control*, New York: IEEE Press (1990).
18. A.J. Laub "Computation of Balancing Transformation," *Proc. 1980 JACC*, San Francisco, CA: Session FA8-E (1980).
19. R.V. Patel and P. Misra, "Numerical Algorithms for Eigenvalue Assignment by State Feedback," "*Proc. IEEE*, vol. 72, pp. 1755-1764 (1984).
20. J.L. Melsa and S.K. Jones, *Computer Program for Computational Assistance in the Study of Linear Control Theory*, New York: McGraw Hill (1973).
21. P. Agathoklis, F.E. Cellier, M. Djoudjevic, P.O. Grepper, and F.J. Kraus, "INTOPS, Educational Aspects of Using Computer-Aided Design in Automatic Control," in *Proceedings of the IFAC Symposium on Computer-Aided Design*, Zürich, Switzerland, August 29-31, Oxford: Pergamon Press, pp. 441-446 (1979).
22. C. Schmid, *KEDDC, User's Manual and Programmer's Manual*, Lehrstuhl fur Elektrische Steuerung and Regulung, Ruhr University Bochum, Germany (1979).
23. C. Schmid, KEDDC-A Computer-Aided Analysis and Design Package for Control Systems, in: *Computer-Aided Control Systems Engineering* (M. Jamshidi and C.J. Herget, eds.), Amsterdam: North Holland, pp. 159-180 (1985).
24. ANSI, *American National Standard for Information Systems, Computer Graphics Graphical Kernel System (GKS)*, Functional Description (ANSI X3. 124-1985) and FORTRAN Binding (ANSI X 3.124.1-1985) (1985).
25. S.C. Shah, M.A. Floyd, and L.L. Lehman, "MATRIX: Control Design and Model Building CAE Capability," in *Computer-Aided Control Systems Engineering*, M. Jamshidi and C.J. Herget (eds.), Amsterdam: North-Holland, pp. 181-207 (1985).

26. P.M. Thompson and S.A. Wolfe, *User's Guide to Program CC*, California Institute of Technology, Pasadena, CA (1983).
27. J.N. Little and C. Moler, *PC-MATLAB, User's Guide, Reference Guide, and On-Line HELP, Brouse and Demonstrations*, The Math Works, Inc., 158 Woodland St., Sherborn, MA 01770 (1986).
28. J. Robinson and R. Kisner, "An Intelligent Dynamic Simulation Environment: An Object-Oriented Approach," *Proc. IEEE Symp. Intelligent Control*, Arlington, VA (1988).
29. S.R. O'Neil and M. Jamshidi, "ROBOT_S: An Interactive Robot Simulation Language," *Robotics*, Vol. 4, No. 3, pp. 245-256 (1988).
30. C.M. Rimvall, *IMPACT, Interactive Mathematical Program for Automatic Control*, Swiss Federal Institute of Technology, ETH-Zentrum, CH-8092 Zurich, Switzerland, p. 208 (1983).
31. M. Jamshidi and C.J. Herget, *Advances in Computer-Aided Control Systems Engineering*, (eds.) Amsterdam: North Holland, forthcoming (1993).
32. J.N. Little, Emani-Naeini, and S.N. Bangert, "CTRL-C and Matrix Environments for the Computer-Aided Design of Control Systems," in *Computer-Aided Control Systems Engineering*, M. Jamshidi and C.J. Herget (eds.), Amsterdam: North Holland (1985).
35. M. Jamshidi and G.L. Schotik, "CONTROL.lab-A CAD Language for Systems Engineering," *Tech. Report 87-01*, CAD Lab. Systems/Robotics, EECE Dept., University of New Mexico, Albuquerque, NM (1987).
36. ISI-Integrated Systems, Inc., *MATRIXx User's Guide, MATRIXx Reference Guide, MATRIXx Training Guide, Command Summary*, and *On-Line Help*, Integrated Systems, Inc., Palo Alto, CA (1984).
37. M. Jamshidi and G.L. Schotik, "CONTROL.lab User's Guide-A CAD Language for Multivariable Control Systems," CAD Lab. Systems/Robotics, *Tech. Report LCAD 86-04*, EECS Dept., University of New Mexico, Albuquerque, NM (1987).

38. Tektronix, Inc., *PLOT 10 Advanced Graphics II User's Manual*, Manual Part No. 070-2244-00 (Rev. Ed) (1982).
39. L. Ljung, *System-Identification Toolbox*, User's Manual, The Mathworks, Inc., Sherborn, MA (1988).
39. L. Ljung, *System Identification: Theory for the User*, Englewood Cliffs, NJ: Prentice Hall (1987).
41. M. Jamshidi, N. Miller, J.A. Meinhardt, and R.A. Carreras, "Dynamic Modeling of a Phased-Array Telescope," *Proc. SPIE Symposium on Advanced Technology Optical Telescopes IV*, vol. 1236, pp. 433-444 (1990).
42. R.Y. Chiang, and M.G. Safanov, *Robust-Control Toolbox*, User's Manual, South Natick, MA: The Mathworks, Inc. (1988).
43. G. Milne, *State-Space Identification Tool*, User's Manual, The Mathworks, Inc., Sherborn, MA (1988).
44. M. Jamshidi, *Large-Scale Systems-Modeling and Control*, New York: North Holland, Elsevier (1983).
45. D. Xue, A. Goucem, and D.P. Atherton, "A Menu Interface to PC-MATLAB for a First Course on Control," *Int. J. Electrical Engineering Education* (forthcoming, 1992).
46. W. Honey, "Robotics Toolbox for the MATLAB Matrix Manipulation Program," *Proc. ANS Topical Meeting on Robotics and Remote Systems*, M. Jamshidi and P. Eicker (eds.), Albuquerque, NM (1991).
47. W. Honey and M. Jamshidi, "ROBO-SIM: A Robotics Simulation Environment on a Personal Computer," *Proc. IFAC Conference on Robots*, Vienna, Austria (1991).
48. M. Jamshidi and M. Malek-Zavarei, 1986, *Linear Control Systems-A Computer-Aided Approach*, Oxford, England: Pergamon Press, Ltd.
49. S. Bingulac, "CAD Package L-A-S: A Resource and Educational Tool in Systems and Control," *Proc. 12th Southeastern Symp. on System Theory*, Charlotte, NC, March (1988).

50. H. Elmqvist, SIMNON-An Interactive Simulation Language for Nonlinear Systems, in: Proceedings of the International Symposium SIMULATION '77, Montreux, Switzerland, M. Hamza, ed., Anaheim, CA: Acta Press, pp. 85-90 (1977).
51. B. Lawver, EAGLES, An Interactive Environment and Program Development Tool, Personal Communication, Dynamics and Controls Group, Lawrence Livermore National Laboratory, University of California, Livermore, CA (1985).
52. Symbolics, Inc., *MACSYMA Reference Manual*, Version 10, MIT and Symbolics, Inc., Cambridge, Ma (1983).
53. Rand Corporation 87, *REDUCE User's Manual*, Version 3.3, Publication CP78, The Rand Corp., Santa Monica, CA.
54. Microsoft, Inc., *MuMath Symbolic Mathematics Package*, Soft Warehouse, Honolulu, HI (1983).

Implicit Model Techniques and Their Application to LQ Adaptive Control

Giuseppe Casalino
Antonella Ferrara
Riccardo Minciardi

Department of Communications, Computer and System Sciences
DIST-University of Genoa, Via Opera Pia 13, 16145 Genova, Italy

Thomas Parisini

Department of Electrical, Electronic and Computer Engineering
DEEI–University of Trieste
Via Valerio 10, 34175 Trieste, Italy

I. INTRODUCTION

The term self–tuning control was introduced by Åström and Wittenmark [1] to denote (indirect) adaptive control schemes based on the recursive identification of some model of a system, followed by the application of a control law by the use of the identified system model. Åström and Wittenmark considered a special case (corresponding to minimum–variance control), which was extremely attractive due to the very easy controller design procedure and to the possibility of identifying an ARX model even in the presence of a true system with an ARMAX structure.

Most of the subsequent research on adaptive algorithms of the self–tuning type [2],[3],[4],[5], [6],[7],[8] was focused on the goal of extending the results of Åström and Wittenmark to more complex control objectives and to multivariable systems. Pole–assignment design objectives were also considered [7],[8].

Other works (see, for instance [9],[10]) considered adaptive algorithms based on multi-step or infinite-horizon LQ cost functionals (see also the recent book [11] for a general treatment of this topic).

In [12], great attention was paid to the second important feature of the self-tuning class of algorithms, namely, the possibility of using and identifying ARX models instead of ARMAX ones (in general, the latter are assumed to represent, true system models). More specifically the term "implicit" was used to denote an ARX model of an ARMAX system which can be used to correctly predict the system output under certain closed-loop conditions. The main reason for such implicit models derives of course from the fact that they can be identified by means of simple least-squares methods, whereas ARMAX models require more complicated identification techniques. The results presented in [12] provided some conditions for the existence of such implicit models in connection with the application of control laws derived from pole-placement or LQ design criteria. Such conditions are recalled and discussed in the present contribution.

Clearly, the interest in ARX implicit models in the context of adaptive control is related to the possibility of using them instead of true system (ARMAX) models, still obtaining the optimal (or desired) regulator according to the controller design criterion adopted. To this end, it was shown [12] that, as far as LQ regulator design is concerned, only in the case of a finite-horizon (and *not* in the case of an infinite-horizon) control objective is it possible to define correctly an adaptive control algorithm based on the identification of a *single* ARX implicit model. In addition, the necessity was demonstrated for a large amount of a-priori information (increasing with the control horizon) about the true system in order to structure the algorithm properly.

Subsequent papers [13],[14] showed that the necessity for a-priori information can be bypassed whenever control laws including the presence of dither noise are used and (slightly) more complex ARX models are identified. On the basis of such models, Mosca and Zappa [15] developed an approach to multistep LQ adaptive control, making use of the recursive identification of a set of models. Mosca and Lemos [16] provided a control algorithm for infinite-horizon LQ adaptive control, still based on the recursive identification of a set of such models.

In all these algorithms, the use of the information provided by the identified implicit model (or a set of implicit models) is conceptually the same as in the papers by Åström and Wittenmark and in subsequent works. More precisely, the identified model is treated as if it were the true system model (in accordance with the so-called certainty equivalence principle) and, at every adaptation step, the control law is completely redesigned.

Subsequently, in [17],[18] attention was focused on a new class of adaptive algorithms, still based on the identification of a *single* ARX implicit model but characterized by the different use of the information provided by such a model. Actually, the basic idea of using the identified model as if it were the true system is retained, but, in this new scheme, the pre-existing control law is "adjusted", instead of being completely redesigned at each adaptation step. More specifically, each adaptation consists in the determination of the "variation" in the pre-existing control law, and this variation is computed on the basis of a such control law and of the currently identified prediction model. Accordingly, the term "variational" was used to denote this adaptive control scheme, which can be based on finite– as well as infinite–horizon quadratic optimization criteria. In the present work, the basic definition of such an algorithm is provided and the possible convergence points (which will also be called "equilibrium points") are discussed. In this respect, two important results will be reported. First, it will be recalled that the choice of a suitable structure for the algorithm allows one to ensure the existence of a unique "whitening" (i.e., with a white prediction error sequence) equilibrium point corresponding to the optimal control law. Secondly, simple sufficient conditions will be defined for a generic convergence point of the algorithm to correspond to this unique whitening equilibrium point. Due to such conditions, a modification to the basic algorithm is introduced for the purpose of preventing the possibility of a convergence point where the aforesaid conditions are not satisfied. Therefore, under suitable hypotheses, the new algorithm is characterized by the so-called "self-tuning property". In other words, whenever a convergence point is reached, this necessarily corresponds to the optimal control law.

The contribution is organized as follows: in the next section some basic concepts and results about implicit models are reported. In Section III, conditions for the existence of implicit models in connection with LQ adaptive criteria are given. In Section IV, the basic variational adaptive algorithm is presented, while in Section V some modifications to the above algorithm in order to guarantee the enjoyment of the self–tuning property are discussed. Finally, Section VI reports some simulation results showing the effectiveness of the proposed approach.

II. BASIC CONCEPTS AND RESULTS

In this section, the concept of implicit model will be introduced and conditions for the existence of implicit models will be provided. The definition of implicit model that will be used in this contribution is consistent with the definition of "extended" implicit models introduced first in [13],[14].

Also results already provided in previous papers are reported here, for the reader's convenience.

Let us refer to the following ARMAX S.I.S.O. system:

$$A(q^{-1})\, y_i = B(q^{-1})\, u_i + C(q^{-1})\, e_i \tag{1}$$

where

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_n q^{-n} \tag{2a}$$

$$B(q^{-1}) = b_1 q^{-1} + b_2 q^{-2} + \cdots + b_n q^{-n} \tag{2b}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + \cdots + c_n q^{-n} \tag{2c}$$

with $\partial(A, B, C) = n$ (i.e., at least one among $\{a_n, b_n, c_n\}$ is $\neq 0$); $A(z)$, $B(z)$ with no unstable common factors; $C(z)$ with zeros outside the unit circle and $\{e_i\}$ stationary white sequence. All the polynomials are in the delay operator q^{-1} We assume that system (1) is governed by the control law:

$$R(q^{-1})\, u_i = P(q^{-1})\, y_i + T(q^{-1})\, \eta_i \tag{3}$$

where

$$R(q^{-1}) = 1 + r_1 q^{-1} + r_2 q^{-2} + \cdots + r_k q^{-k} \tag{4a}$$

$$P(q^{-1}) = p_0 + p_1 q^{-1} + p_2 q^{-2} + \cdots + p_k q^{-k} \tag{4b}$$

$$T(q^{-1}) = 1 + t_1 q^{-1} + t_2 q^{-2} + \cdots + t_k q^{-k} \tag{4c}$$

being $\partial(R, P, T) = k$ and $\{\eta_i\}$ an arbitrary sequence, acting as a "dither noise". The closed-loop system (1),(3) is supposed to be asymptotically stable.

With the above described conditions relating to the system and the feedback law, the following definition is given.

Definition 2.1. *A model*

$$\mathcal{A}(q^{-1})\, y_i = \mathcal{B}(q^{-1})\, u_i + \mathcal{D}(q^{-1})\, \eta_i + e_i \tag{5}$$

where

$$\mathcal{A}(q^{-1}) = 1 + \alpha_1 q^{-1} + \alpha_2 q^{-2} + \cdots + \alpha_h q^{-h} \tag{6a}$$

$$\mathcal{B}(q^{-1}) = \beta_1 q^{-1} + \beta_2 q^{-2} + \cdots + \beta_h q^{-h} \tag{6b}$$

$$\mathcal{D}(q^{-1}) = \delta_1 q^{-1} + \delta_2 q^{-2} + \cdots + \delta_h q^{-h} \tag{6c}$$

is an **implicit model** *for the closed-loop system (1),(3), if, for any couple of realizations of the sequences* $\{e_i\}$ and $\{\eta_i\}$, *the sequence* $\{y_i\}$ *given by (5) coincides asymptotically (i.e., in the stochastic steady state) with the one given by (1) when controlled by (3).*

□

As mentioned in the Introduction, the use of a model like (5) in an adaptive control scheme is compatible with the use of Recursive Least Squares identification. This is due to the perfect knowledge of the samples η_i .

The following theorem provides necessary and sufficient conditions for the existence of implicit models.

Theorem 2.1. *The asymptotically stable closed-loop system (1),(3) admits implicit models if and only if the following conditions hold:*

i) the closed-loop characteristic polynomial has the structure

$$AR - BP = QC \tag{7}$$

(this relation implicitly defines polynomial $Q(q^{-1})$*)*

ii)

$$\gcd(R, P) = \gcd(R, P, Q) \tag{8}$$

iii) polynomial T *can be factorized as*

$$T = SC^\circ \tag{9}$$

where

$$(B^\circ, C^\circ) \triangleq \frac{(B, C)}{\gcd(B, C)}$$

and $S(q^{-1})$ *is a monic polynomial.*

iv) at least in correspondence with one pair $(\mathcal{A}, \mathcal{B})$ *solving the Bezoutian equation*

$$\mathcal{A}R - \mathcal{B}P = Q \tag{10}$$

(which has surely solutions whenever condition ii) holds)

$$S(B^\circ - C^\circ \mathcal{B}) \quad \text{is a multiple of } R \tag{11}$$

Besides, condition (iii) can be substituted by the following *iii bis) polynomial $T(q^{-1})$ can be factorized as*

$$T = \bar{S}\bar{C}^{\circ} \tag{12}$$

where

$$(A^{\circ}, C^{\circ}) \triangleq \frac{(A, C)}{\gcd(A, C)}$$

and $\bar{S}(q^{-1})$ is a monic polynomial.

Similarly condition iv) can be substituted by

iv bis) at least in correspondence with one pair $(\mathcal{A}, \mathcal{B})$ solving the Bezoutian equation (10),

$$S(A^{\circ} - C^{\circ}\mathcal{A}) \quad \text{is a multiple of } P \tag{13}$$

Moreover, in case the above necessary and sufficient conditions hold, every triple $(\mathcal{A}, \mathcal{B}, \mathcal{D})$, where $(\mathcal{A}, \mathcal{B})$ satisfy (10) and (11), or (13), and $\mathcal{D}$ is given by

$$\mathcal{D} = \frac{S(B^{\circ} - C^{\circ}\mathcal{B})}{R} \tag{14}$$

or, equivalently, by

$$\mathcal{D} = \frac{S(A^{\circ} - C^{\circ}\mathcal{A})}{P} \tag{15}$$

is an implicit model, and no other implicit model exists.

□

Proof. See the Appendix.

The conditions of Theorem 2.1 are both necessary and sufficient for the existence of implicit models for an asymptotically stable closed-loop system having the structure given by (1),(3). Unfortunately, condition iv), or equivalently condition iv bis), is not so easy to verify on the basis of the pure knowledge of (A, B, C) and (R, P, T). Thus, it is worth providing simpler sufficient conditions, directly based on (A, B, C) and (R, P, T).

Theorem 2.2. *The asymptotically stable closed-loop system (1),(3) admits implicit models if the following conditions hold:*

i)– ii) as in Theorem 2.1.

iii ter) $T(q^{-1})$ *is a multiple of* $\gcd(R,P)\,C_0$ *, where* $C_0 \triangleq \dfrac{C}{\gcd(A,B,C)}$

Proof. See the Appendix.

The following theorem provides a characterization of the class of implicit models associated to a given ARMAX systems (1) and with a given regulator (3).

Theorem 2.3. *Assume that the sufficient conditions in Theorem 2.2 hold (which ensures that the (stable) closed loop system (1),(3) admits implicit models). Define* $\bar{k} \triangleq \partial(R,P)$, $j \triangleq \partial(Q)$. *Then the following properties hold.*

i) The set $\mathcal{I}_h$ *of the implicit models characterized by* $\partial(\mathcal{A},\mathcal{B}) \leq h$ *is non-empty if* $h \geq h_0 \triangleq \max(\bar{k}, j-\bar{k})$.

ii) For any h *satisfying point i), the set* $\mathcal{I}_h$ *is a* $(h-\bar{k})$*-dimensional linear manifold. In particular, if* $\bar{k} > j-\bar{k}$ *(i.e.,* $h_0 = \bar{k}$ *),* $\mathcal{I}_h$ *contains a single implicit model only for* $h = h_0 = \bar{k}$.

iii) If $\bar{k} < j-\bar{k}$ *(i.e.,* $h_0 = j-\bar{k}$ *) it is never possible to choose* h *such that* $\mathcal{I}_h$ *is composed by a single implicit model.*

□

Proof. First, note that, if Assumption iii ter) of Theorem 2.2 holds, then, in connection to every $(\mathcal{A},\mathcal{B})$ satisfying (10), one and only one polynomial $\mathcal{D}$ can be found via (14) or (15). This allows us to focus the attention only on pairs $(\mathcal{A},\mathcal{B})$ and not on triples $(\mathcal{A},\mathcal{B},\mathcal{D})$.

Let us then prove the above properties separately.

Property i). Observe that, for a solution of (10), characterized by $\partial(\mathcal{A},\mathcal{B}) \leq h$ to exist, it must be necessarily

$$h+\bar{k} \geq j \implies h \geq j-\bar{k} \tag{16}$$

Moreover, with h satisfying (16), equating both sides of (10) term by term, one gets $h+\bar{k}$ linear equations in the $2h$ unknowns represented by the coefficients of $\mathcal{A},\mathcal{B}$. Due to the fact that R and P are coprime, the coefficient matrix of the resulting set of equations has full rank. Then, a sufficient condition to find at least one solution is to set the maximum admissible order h such that:

$$h+\bar{k} \leq 2h \implies h \geq \bar{k} \tag{17}$$

Putting (16) and (17) together, finally gives that the maximum order h should always be chosen satisfying the condition

$$h \geq h_0 \triangleq \max(\bar{k}, j - \bar{k}) \tag{18}$$

Note that (18) is a sufficient, but not necessary condition, since, in some particular instances, a solution might exist even with h not satisfying (17).

Properties ii) and iii). These follow directly from elementary linear algebra considerations.

∎

Finally, the following theorem provides a useful characterization of a class of implicit models.

Theorem 2.4. *Under the conditions of Theorem 2.2, the implicit models satisfying one of the following three conditions also satisfy the other two:*

i) The first $(m+1)$ coefficients of $\mathcal{A}$ are equal to the first $(m+1)$ coefficients of the expansion of A/C;

ii) the first m coefficients of $\mathcal{B}$ are equal to the first m coefficients of the expansion of B/C;

iii) the first m coefficients of $\mathcal{D}$ are equal to zero.

□

Proof. The proof of the mutual implication of (i) and (iii) derives from (15). Equivalently the proof of the mutual implication between (ii) and (iii) derives from (14). The equivalence between (i) and (ii) is thus also proved.

∎

III. EXISTENCE OF IMPLICIT MODELS IN PRESENCE OF REGULATORS DERIVING FROM QUADRATIC OPTIMIZATION

Up to this point, no specification about the type of control law (3) governing system (1) has been made. Now, we investigate the question of existence of implicit models in connection with the presence of a regulator (3) coming out from the optimization of a quadratic cost functional. More specifically, we address a general quadratic optimization problem (GQOP), consisting in the minimization (at each time instant i, and in the receding-horizon sense) of the cost functional

$$E\left[\sum_{t=i}^{i+m-1} (y_{t+1}^2 + p\,u_t^2) \mid I_i\right] \quad (p > 0) \tag{19a}$$

or its infinite horizon counterpart

$$\lim_{m\to\infty} \frac{1}{m} E\left[\sum_{t=i}^{i+m-1} (y_{t+1}^2 + p\,u_t^2) \mid I_i\right] \quad (p>0) \tag{19b}$$

with respect to strategies of the form

$$u_i = \overline{u}_i + \eta_i \tag{20}$$

where

$$\overline{u}_i = \gamma[I_i]$$

$$I_i \triangleq \{y_i, y_{i-1}, \ldots, u_{i-1}, u_{i-2}, \ldots, \eta_{i-1}, \eta_{i-2}, \ldots\}$$

and $\{\eta_i\}$ stationary white sequence

Then, it is a well known result that the stationary optimal strategy solving this problem can be put in the polynomial form

$$R(q^{-1})\,u_i = P(q^{-1})\,y_i + C(q^{-1})\eta_i \tag{21}$$

where polynomials $R(q^{-1})$ and $P(q^{-1})$ have the structure given by (4) with $k \le n$ and $C(q^{-1})$ is the same as in (1). Moreover $C(q^{-1})$ turns out to be a factor of the closed-loop characteristic polynomial, i.e., a polynomial $Q(q^{-1})$ exists such that $A\,R - B\,P = Q\,C$.

Theorem 3.1. *Let system (1) be governed by a control law of type (21), which is assumed to solve the GQOP (minimization of (19a) or (19b)). Then, the following set of conditions are sufficient in order to ensure that implicit models exist:*

a) the regulator (R, P, T) stabilizes the system (A, B, C);

b) gcd (R, P) *is a common factor of polynomial $Q(q^{-1})$, above defined, and of polynomial* gcd (A, B, C).

□

Proof. As previously mentioned, condition *i*) of Theorem 2.2 is always fulfilled for the class of control laws considered. Moreover, $T(q^{-1}) = C(q^{-1})$. Thus, it is apparent that *b)* implies that also condition *ii*) and *iii ter*) are satisfied.

■

Note that the fulfilment of condition a) is generally verified in the case of infinite horizon optimization. The same can be said also in the case of finite horizon optimization, at least for "large enough" values of m. As to condition b), since it involves only common factors of R, P (which are generally absent), it can be considered as generally verified. Thus, on the whole, in connection with controllers deriving from the solution of a GQOP, we can say that, in general, implicit models exist.

In addition, it can be easily shown that if implicit models exist in connection with a regulator (R, P, T), then they exist also in presence of any stabilizing regulator $(R\Gamma, P\Gamma, T\Gamma)$, where Γ is an arbitrary stable polynomial.

IV. APPLICATION OF IMPLICIT MODELS TO LQ ADAPTIVE OPTIMIZATION: THE VARIATIONAL ALGORITHM

In this section, a possible way of utilizing the concepts and properties of implicit models to develop adaptive control algorithms based on quadratic optimization criteria will be presented. Actually, Casalino *et al.* [13, 14] proposed a possible way of making use of such concepts within an adaptive control scheme aiming at the optimization of an m-step-ahead quadratic cost. Such a scheme was essentially based on the parallel identification of m implicit models of different orders. For this reason, it was not possible to face the problem of infinite-horizon quadratic optimization. Thus, one was led to search for a different way of using the information provided by an implicit model.

In order to describe this approach, it is advisable first to define a pair of control problems and then to use their equivalence. In this connection, let us assume that: (a) system (1) has been controlled by a stabilizing regulator of type (3) up to time instant $(i - 1)$, being $\{\eta_i\}$ an arbitrary stochastic sequence; (b) system (1) and controller (3) are such that implicit models exist. Then, let us consider the following problems.

Problem 4.1. *Find the control strategies* $u_t = \overline{u}_t + \eta_t$, $t \geq i$, *with* $\overline{u}_t = \gamma_t[I_t]$ *and* $\{\eta_t\}$ *stationary white sequence, which minimize cost functional (19a), or its infinite-horizon counterpart (19b).*

□

Problem 4.2. *Find the control strategies giving the "optimal variations"* $\delta u_t = \widetilde{\gamma}_t[\widetilde{I}_t]$, $t \geq i$, *such that the control strategies given by*

$$u_t = \left[1 - R(q^{-1})\right] \ u_t + P(q^{-1})\, y_t + T(q^{-1})\,(\eta_t + \delta u_t)\, , \, t \geq i \qquad (22)$$

minimize the cost functional (19a) or (19b), where $\{\eta_t\}$ *has the same meaning as above,* $\delta u_t = 0$, $t < i$, *and*

$$\widetilde{I}_t \triangleq \{y_t, y_{t-1}, \ldots, u_{t-1}, u_{t-2}, \ldots, \eta_{t-1}, \eta_{t-2}, \ldots, \delta u_{t-1}, \delta u_{t-2}, \ldots\}$$

The two above introduced problems are *equivalent*, in the sense that they give *the same control actions* for $t \geq i$. To understand this, it is sufficient to note that the first coefficient in polynomial $T(q^{-1})$ is 1 (i.e., it is different from zero). Thus, it is possible to represent *any* present or future (with respect to time instant i) control action u_t, $t \geq i$, in the form of (22). Then, the two optimality problems must yield the same result, in particular as regards the control action u_i.

Furtherly, we observe that, thanks to the assumption of whiteness of $\{\eta_t\}$ the same solutions of Problem 4.1 and Problem 4.2 (as regards the dependence of strategies γ_t and $\widetilde{\gamma}_t$ on the other components of the information sets I_t and $\widetilde{I}_t$, respectively) can be found even if we assume $\eta_t \equiv 0 \quad \text{for} \quad t \geq i$.

Thus, the controller structure (22) becomes simply

$$u_t = \left[1 - R(q^{-1})\right] u_t + P(q^{-1}) y_t + T(q^{-1}) \eta_t \qquad , \quad t \geq i$$

where we have renamed δu_t as η_t, for $t \geq i$, thus interpreting now η_t as an arbitrary sequence for $t \leq i-1$ and a manipulable input for $t \geq i$.

Now, it can be observed that the absolute generality of the hypotheses about the sequence $\{\eta_t\}$ in the previous section allows us to state that the same implicit models, which we assume to exist for $t < i$, exist even for $t \geq i$ and can be used to predict the output y_t, $t \geq i$.

Thus, Problem 4.2 can be correctly stated and solved using, instead of the true system model, the coupled equations of the pre-existing regulator and one of the associated implicit models, namely

$$\mathcal{A}(q^{-1}) y_t = \mathcal{B}(q^{-1}) u_t + \mathcal{D}(q^{-1}) \eta_t + e_t \tag{23}$$

$$R(q^{-1}) u_t = P(q^{-1}) y_t + T(q^{-1}) \eta_t \tag{24}$$

where $\partial(\mathcal{A}, \mathcal{B}, \mathcal{D}) = h$ and $\{\eta_t\}$ has the twofold meaning above specified.

The solution of Problem 4.2 can be obtained by referring to the following (non-minimal, for the sake of simplicity) state space representation of the coupled equations (23),(24), and considering η_t as the input and y_t, u_t as the outputs:

$$\boldsymbol{x}_{t+1} = F\,\boldsymbol{x}_t + \boldsymbol{g}\,\eta_t + \boldsymbol{k}\,e_t \tag{25a}$$

$$y_{t+1} = \boldsymbol{h}^T \boldsymbol{x}_{t+1} + e_{t+1} \tag{25b}$$

$$u_t = \boldsymbol{s}^T \boldsymbol{x}_{t+1} \tag{25c}$$

where (supposing $k \leq h$)

$$\boldsymbol{x}_t = \mathrm{col}\left[y_{t-1}, \ldots, y_{t-h}, u_{t-1}, \ldots, u_{t-h}, \eta_{t-1}, \ldots, \eta_{t-h}\right] \tag{26}$$

$$F = \left[\begin{array}{ccc}
\overline{1-\mathcal{A}} & \overline{\mathcal{B}} & \overline{\mathcal{D}} \\
1\,0\ldots\ldots0 & 0\ldots\ldots0\,0 & 0\ldots\ldots0\,0 \\
\ldots\ldots & \ldots & \ldots \\
0\ldots\ldots1\,0 & 0\ldots\ldots0\,0 & 0\ldots\ldots0\,0 \\
 & & \\
\overline{(P-p_0)+p_0(1-\mathcal{A})} & \overline{(1-R)+p_0\,\mathcal{B}} & \overline{(T-1)+p_0\,\mathcal{D}} \\
0\ldots\ldots0\,0 & 1\,0\ldots\ldots0 & 0\ldots\ldots0\,0 \\
\ldots\ldots & \ldots & \ldots \\
0\ldots\ldots0\,0 & 0\ldots\ldots1\,0 & 0\ldots\ldots0\,0 \\
 & & \\
0\ldots\ldots0\,0 & 0\ldots\ldots0\,0 & 0\ldots\ldots0\,0 \\
0\ldots\ldots0\,0 & 0\ldots\ldots0\,0 & 1\,0\ldots\ldots0 \\
\ldots\ldots & \ldots & \ldots \\
0\ldots\ldots0\,0 & 0\ldots\ldots0\,0 & 0\ldots\ldots1\,0
\end{array}\right] \tag{27a}$$

($\overline{Z}$ being defined as a row vector collecting the ordered coefficients of a generic polynomial $Z(q^{-1})$)

$$\boldsymbol{g}^T = [0\ldots0\,|\,1\,0\ldots0\,|\,1\,0\ldots0] \tag{27b}$$

$$\boldsymbol{k}^T = [1\,0\ldots0\,|\,p_o\,0\ldots0\,|\,0\ldots0] \tag{27c}$$

$$\boldsymbol{h}^T = [\,\overline{1-\mathcal{A}}\,\,|\,\overline{\mathcal{B}}\,|\,\overline{\mathcal{D}}\,] \tag{27d}$$

$$\boldsymbol{s}^T = [0\ldots0\,|\,1\,0\ldots0\,|\,0\ldots0] \tag{27e}$$

Then the finite horizon quadratic cost (19a) takes on the equivalent form

$$E\left[\sum_{t=i}^{i+m-1} \|\boldsymbol{x}_{t+1}\|^2{}_W \mid I_i\right] \tag{28}$$

and a similar form is assumed by its infinite horizon counterpart (19b), with

$$W \triangleq p\,\boldsymbol{s}\,\boldsymbol{s}^T + \boldsymbol{h}\,\boldsymbol{h}^T \tag{29}$$

The optimal control strategy can be obtained by applying dynamic programming to this state space model.

Theorem 4.1. *The solution of Problem 4.2 for time instant i can be obtained by means of the following procedure. First, iterate $(m-1)$ times (in case of m-step quadratic cost) , or till convergence (in case of infinite horizon quadratic cost) the matrix Riccati equation*

$$\begin{cases} M_{k+1} = F^T \left[M_k - M_k \boldsymbol{g} \left(\boldsymbol{g}^T M_k \boldsymbol{g} \right)^{-1} \boldsymbol{g}^T M_k \right] F + W \\ \qquad k = 0, 1, \ldots \\ M_0 = W \end{cases} \tag{30}$$

(where the scalar term to be inverted is certainly nonzero at any iteration, because of the assumption $p > 0$). The convergence of (30) is always assured. Then, the matrix M^* , computed in the above specified way, is used to compute the optimal control variation

$$\delta u_i{}^{\circ} = -\left(\boldsymbol{g}^T M^* \boldsymbol{g} \right)^{-1} \boldsymbol{g}^T M^* \left[(F - \boldsymbol{k}\boldsymbol{h}^T) \boldsymbol{x}_i + \boldsymbol{k}\, y_i \right] \tag{31}$$

□

Proof. At this point, the theorem needs to be proved only as regards: *i)* the possibility of inverting the scalar term, *ii)* the convergence of the matrix Riccati equation. As far as the first issue is concerned, we recall that, at the first iteration, $M_k = W$. Then, using (29), one has the following expression for the term to be inverted at the first iteration:

$$\begin{aligned} \boldsymbol{g}^T W \boldsymbol{g} &= \boldsymbol{g}^T \left[p \boldsymbol{s}\boldsymbol{s}^T + \boldsymbol{h}\boldsymbol{h}^T \right] \boldsymbol{g} \\ &= p (\boldsymbol{g}^T \boldsymbol{s}) \cdot (\boldsymbol{s}^T \boldsymbol{g}) + (\boldsymbol{g}^T \boldsymbol{h}) \cdot (\boldsymbol{h}^T \boldsymbol{g}) \end{aligned} \tag{32}$$

It can be observed that, due to the structure of $\boldsymbol{g}$ and $\boldsymbol{s}$, the first term in the r.h.s. is equal to p (> 0), whereas the second term is certainly non–negative. Thus, the term to be inverted at the first iteration is certainly nonzero. The term to be inverted at any generic subsequent iteration is given by

$$\boldsymbol{g}^T (\widehat{M}_{k+1} + W) \boldsymbol{g} = \boldsymbol{g}^T \widehat{M}_{k+1} \boldsymbol{g} + \boldsymbol{g}^T W \boldsymbol{g} \tag{33}$$

where $\widehat{M}_{k+1}$ represents the first term on the r.h.s. of (30). Then, since $\widehat{M}_{k+1}$ is a positive semidefinite matrix, the term given by (33) is still certainly nonzero. Finally, the Riccati equation (30) converges, according to the assumption about the stability of the closed–loop systems [19].

■

We have shown that solving Problem 4.2 is equivalent to solving Problem 4.1. Moreover, Theorem 4.1 shows a way to compute the solution of Problem 4.2, as regards the control action to be applied at time instant i. Then, we can think of coupling a procedure for the recursive identification of an implicit model with an adaptation step consisting not in the complete redesign of the control law, but in *an adjustement* of the pre-existing one, on the basis of the expression of the optimal control variation at time instant i. More formally, we can introduce the following adaptive scheme.

Algorithm 4.1: Variational Adaptive Control Algorithm (for Quadratic Optimization over a finite or an infinite horizon). *At every iteration i, the following operations take place:*

(a) updating the parameter estimates of an implicit model via Recursive Least Squares, thus obtaining $(\widehat{\mathcal{A}}_i, \widehat{\mathcal{B}}_i, \widehat{\mathcal{D}}_i)$ (of fixed maximum order h);

(b) determination of matrix M_i^*, as shown in Theorem 4.1, on the basis of a state-space representation with structure (25)–(27) of the coupled system (23),(24), where the polynomials are given by $(\widehat{\mathcal{A}}_i, \widehat{\mathcal{B}}_i, \widehat{\mathcal{D}}_i)$, $(R_{i-1}, P_{i-1}, T_{i-1})$ (the order k of the previously acting regulator is assumed to be $\leq h$);

(c) computation of the control variation

$$\delta u_i = -(\boldsymbol{g}^T M_i^* \boldsymbol{g})^{-1} \boldsymbol{g}^T M_i^* \left[(F_i - \boldsymbol{k}_i \boldsymbol{h}_i^T) \boldsymbol{x}_i + \boldsymbol{k}_i y_i \right] \tag{34}$$

(the dependence of matrices and vectors on the iteration number is put into evidence) and application of the control action

$$u_i = \overline{u}_i + \delta u_i + \eta_i \tag{35}$$

where $\{\eta_i\}$ is a white stationary sequence and $\overline{u}_i$ is the control action generated by the controller $(R_{i-1}, P_{i-1}, T_{i-1})$ with $\eta_i = 0$;

(d) writing (34) in a polynomial form

$$\delta u_i = \Delta P_i(q^{-1}) y_i - \Delta R_i(q^{-1}) u_i + \Delta T_i(q^{-1}) \eta_i \tag{36}$$

(which implicitly defines polynomials ΔR_i, ΔP_i, ΔT_i) and, on this basis, updating the polynomial expression of the previously acting regulator $(R_{i-1}, P_{i-1}, T_{i-1})$ (by simply adding ΔR_i, ΔP_i, ΔT_i),

thus obtaining polynomials (R_i, P_i, T_i), *again of order* $\leq h$.

In the following, we shall briefly consider some questions related to the equilibrium points of Algorithm 4.1.

Definition 4.1. *A set of polynomials* $\mathcal{A}^*, \mathcal{B}^*, \mathcal{D}^*, R^*, P^*, T^*$ *is said to be an equilibrium point of Algorithm 4.1 if:*

(i) steps (b),(c),(d) of the algorithm, based on the state-space representation of the coupled equations (23),(24) (written in correspondence with the polynomials individuating the equilibrium point), return a regulator characterized by R^*, P^*, T^*;

(ii) in the closed loop system (1),(3), data are generated in such a way that the prediction error

$$\xi_i \triangleq \mathcal{A}^*(q^{-1})\, y_i - \mathcal{B}^*(q^{-1})\, u_i - \mathcal{D}^*(q^{-1})\, \eta_i \tag{37}$$

is uncorrelated with the regression vector (set of the measurements corresponding to the parameters of the implicit model to be identified).

This equilibrium point is called "whitening" if $\{\xi_i\}$ *is a white sequence. Clearly, a whitening equilibrium point corresponds to an implicit model* $(\mathcal{A}^*, \mathcal{B}^*, \mathcal{D}^*)$ *for the closed-loop system characterized by* (A, B, C), (R^*, P^*, T^*).

□

Obviously, in defining the structure of an adaptive control algorithm, we are interested not only in the existence of whitening equilibrium points corresponding to regulators equivalent to the optimal one, but also in making the set of such equilibrium points as reduced as possible, hopefully retaining only equilibrium points corresponding to the (minimal order) optimal control law. To this end, the following result plays a key role.

Theorem 4.2. *Assume that the GQOP referred to thesystem model (1) leads to a regulator* $[R^\circ(q^{-1}), P^\circ(q^{-1}), C(q^{-1}))$ *characterized by the following properties:*

(i) The closed-loop system (1),(3) is asymptotically stable.

(ii) $\gcd(R^\circ, P^\circ) = 1$

(iii) $\partial(R^{\circ}, P^{\circ}) = n$

Moreover, introduce the following specifications in Algorithm 4.1:

- *(a) The maximum order of the implicit model to be identified is fixed to* $(n+1)$ *;*
- *(b) a single coefficient is arbitrarily fixed in* $(\mathcal{A}_i, \mathcal{B}_i, \mathcal{D}_i)$ *;*
- *(c) the last coefficients of polynomials* (R_i, P_i, T_i) *are initialized at the values* $t_0^{(n+1)} = p_0^{(n+1)} = r_0^{(n+1)} = 0$.

Then the algorithm admits a unique whitening equilibrium point which is characterized by the optimal (minimal order) control law.

□

Proof. Let us formally denote by

$$(\Delta R_i, \Delta P_i, \Delta T_i) = \Phi\,(R_{i-1}, P_{i-1}, T_{i-1}, \mathcal{A}_i, \mathcal{B}_i, \mathcal{D}_i)$$

the mapping resulting from operations (b),(c) and (d) in Algorithm 4.1.

Assumptions *i)* and *ii)*, in accordance with Theorem 3.1, ensure that implicit models exist, in connection with the considered regulator. Then, observe that specification (a) defines a parametrized set of implicit models, with one degree of freedom (see Theorem 2.3). Let us remove this degree of freedom by arbitrarily fixing one of the coefficients of the implicit model (as required by (b)). This identifies a single particular implicit model, say $(\mathcal{A}^{\circ}, \mathcal{B}^{\circ}, \mathcal{D}^{\circ})$. The remainder of the proof can be divided into two parts.
Part 1. Let us show that $(\mathcal{A}^{\circ}, \mathcal{B}^{\circ}, \mathcal{D}^{\circ})$ and $(R^{\circ}, P^{\circ}, C)$ together represent a whitening equilibrium point of the algorithm. Then note that condition *ii)* of Definition 4.1 is obviously satisfied. Moreover, it is not difficult to deduce that, due to the initialization specified in (c), the order of the regulator recursively updated by Algorithm 4.1 cannot exceed n. Then, assumptions *ii)* and *iii)* ensure that the optimal control is unique. Hence, by virtue of the uniqueness of the optimal controller we have

$$(\Delta R, \Delta P, \Delta T) = \Phi(R^{\circ}, P^{\circ}, C, \mathcal{A}^{\circ}, \mathcal{B}^{\circ}, \mathcal{D}^{\circ}) = 0 \tag{38}$$

Part 2. Then, let us prove that a given whitening equilibrium point $(R, P, T, \mathcal{A}^{\circ}, \mathcal{B}^{\circ}, \mathcal{D}^{\circ})$ must necessarily correspond to $(R^{\circ}, P^{\circ}, C, \mathcal{A}^{\circ}, \mathcal{B}^{\circ}, \mathcal{D}^{\circ})$.

Obviously, $(\mathcal{A}, \mathcal{B}, \mathcal{D})$ is an implicit model (due to the "whiteness" assumption). Then, (R, P, T) and $(\mathcal{A}, \mathcal{B}, \mathcal{D})$ together represent a regulator and an associated implicit model that satisfy (38). Then, on the basis of the equivalence of Problems 4.1 and 4.2, we can conclude that, as no variation in the control law occurs, (R, P, T) must coincide with the unique optimal controller, i.e., (R°, P°, C); this also implies that $(\mathcal{A}, \mathcal{B}, \mathcal{D})$ coincides with $(\mathcal{A}^\circ, \mathcal{B}^\circ, \mathcal{D}^\circ)$.

V. A VARIATIONAL ADAPTIVE SCHEME ENJOYING THE SELF-TUNING PROPERTY

The results presented show that, under very general hypotheses, it is possible to define in detail the structure of the adaptive control algorithm in such a way that a unique whitening equilibrium point (i.e., a unique whitening possible convergence point) exists, corresponding to the optimal (minimal order) control law. This result, by itself, is not sufficient to assure a good performance of the algorithm, at least because the existence is not excluded of non-whiteninig equilibrium points, i.e., equilibrium points which do not correspond to the presence of an implicit model (and do not yield the optimal, minimal order, control law, nor any equivalent one).

Due to this fact, we are interested in finding conditions over a generic (given) equilibrium point capable of assuring the "whiteness" of such equilibrium point (and hence, in the hypotheses of Theorem 4.2, ensuring that such a point coincides with the unique whiteninig equilibrium point yielding the minimal order optimal control law).

The conditions sought are provided by the following theorem.

Theorem 5.1. *Refer to Algorithm 4.1 (for either infinite horizon or m-step quadratic optimization) with the specifications (1) and (2) introduced in Theorem 4.2. Further assume*

(a) $\gcd(A, B) = 1$ *and* $\partial(A, B) = n$

and that the identification-adaptation algorithm has reached an equilibrium point (see Definition 4.1) $(\mathcal{A}, \mathcal{B}, \mathcal{D}, R, P, T)$ *characterized by the following facts:*

(b) (R, P, T) *is an (asymptotically) stabilizing regulator both for the true controlled system* (A, B, C) *and for the estimated implicit model* $(\mathcal{A}, \mathcal{B}, \mathcal{D})$*;*

(c) $\gcd(R, P) = 1$ *and* $\partial(R, P) = n$

(d) $\partial(\mathcal{A}R - \mathcal{B}P) \leq n$.

Then $(\mathcal{A}, \mathcal{B}, \mathcal{D}, R, P, T)$ *is the unique whitening equilibrium point, and* (R, P, T) *coincides with the minimal order optimal regulator* $(R^{\circ}, P^{\circ}, C)$.

□

Proof. See the Appendix.

Actually, Theorem 5.1 suggests to introduce important modifications in our algorithm. Such modifications are essentially introduced so that, for any equilibrium point of the algorithm, condition (d) of Theorem 5.1 automatically holds. Formally, we can define a "modified" algorithm as follows.

Algorithm 5.1: Modified Variational Adaptive Control Algorithm (for Quadratic Optimization over a finite or an infinite horizon). *Take into account specification (2), introduced in Theorem 4.2, in the initialization of the regulator polynomials. Moreover, initialize arbitrarily polynomial* $Q(q^{-1})$ *(which will be used in the Algorithm) in such a way that* $\partial[Q_0(q^{-1})] \leq n$. *Then, at every iteration* i, *perform the following operations:*

(a) as in Algorithm 4.1, but taking into account specification (1) in Theorem 4.2;

(b1) check the fulfilment of the following condition:

$$\partial\left(\widehat{\mathcal{A}}_i\, R_{i-1} - \widehat{\mathcal{B}}_i\, P_{i-1}\right) \leq n \tag{39}$$

if (39) is satisfied, then set

$$\begin{aligned} R'_{i-1} &= R_{i-1} \\ P'_{i-1} &= P_{i-1} \end{aligned} \tag{40}$$

and go to (b2);

- *otherwise, solve the Bezoutian equation*

$$\widehat{\mathcal{A}}_i\, R'_{i-1} - \widehat{\mathcal{B}}_i\, P'_{i-1} = Q_{i-1} \tag{41}$$

in the n*-th order polynomials* R'_{i-1}, P'_{i-1} *(having the same structure as* R_{i-1}, P_{i-1}, *respectively),* Q_0 *being fixed as above specified, at the first iteration, and*

$$Q_i \triangleq \widehat{\mathcal{A}}_i\, R_i - \widehat{\mathcal{B}}_i\, P_i \tag{42}$$

for $i \geq 1$;

(b2) determine matrix M_i^* *as in Algorithm 4.1, but on the basis of the polynomial triples* $(\widehat{\mathcal{A}}_i, \widehat{\mathcal{B}}_i, \widehat{\mathcal{D}}_i)$, $(R'_{i-1}, P'_{i-1}, T'_{i-1})$;

(c1) compute the law yielding the control variation as in (34), which can be written as in (36);

(c2) check the fulfilment of the following condition

$$\partial\left(\widehat{\mathcal{A}}_i\, \Delta R_i - \widehat{\mathcal{B}}_i\, \Delta P_i\right) \leq n \tag{43}$$

- *in the affirmative case, set*

$$\begin{aligned} \Delta R_i{}' &= \Delta R_i \\ \Delta P_i{}' &= \Delta P_i \end{aligned} \tag{44}$$

and go to (d1);

otherwise solve the following problem:

$$\min \|\boldsymbol{z} - \boldsymbol{z}^\circ\|^2 \tag{45}$$

s.t.

$$\partial(\widehat{\mathcal{A}}_i\, \Delta R_i{}' - \widehat{\mathcal{B}}_i\, \Delta P_i{}') \leq n \tag{46}$$

$\boldsymbol{z}$ *being the vector collecting the ordered coefficients of polynomials* $\Delta R'_i(q^{-1})$, $\Delta P'_i(q^{-1})$ *, namely*

$$\boldsymbol{z} \stackrel{\Delta}{=} [\delta r'_1, \ldots, \delta r'_n, \delta p'_0, \ldots, \delta p'_n]^T \tag{47}$$

and $\boldsymbol{z}^\circ$ *the vector collecting the coefficients of polynomials* $\Delta R_i(q^{-1})$, $\Delta P_i(q^{-1})$;

(d1) update the polynomial expressions of the regulator as follows:

$$R_i = R'_{i-1} + \Delta R_i{}' \tag{48a}$$

$$P_i = P'_{i-1} + \Delta P_i{}' \tag{48b}$$

$$T_i = T_{i-1} + \Delta T_i \tag{48c}$$

(d2) apply the control action given by

$$u_i = \overline{u}_i + \delta u_i + \eta_i \tag{49}$$

where $\{\eta_i\}$ *is a white stationary sequence,* $\overline{u}_i$ *is the control action generated by the controller* $(R'_{i-1}, P'_{i-1}, T'_{i-1})$ *with* $\eta_i \equiv 0$ *, and*

$$\delta u_i \;=\; \Delta P_i'(q^{-1})\, y_i - \Delta R_i'(q^{-1})\, u_i + \Delta T_i(q^{-1})\, \eta_i \tag{50}$$

□

The structure of Algorithm 5.1 needs some comments. First of all, note that, if $\partial(Q_{i-1}) \leq n$, as supposed, then even the polynomial in the l.h.s. of (41) has maximum order n; moreover, step (c2) imposes that the same conditions hold also for the polynomial in (46). Thus, since polynomials (R_i, P_i, T_i) are obtained as indicated in (48), it turns out that even that maximum order of polynomial Q_i defined by (42) is n, i.e., $\partial(Q_i) \leq n$. This actually ensures that at every iteration of the algorithm, $\partial(Q_i) \leq n$.

The rationale for the modifications introduced in the above algorithm, with respect to Algorithm 4.1 is the following. We want to impose that condition (d) in Theorem 5.1 holds in correspondence of every equilibrium point. To this end, we impose this condition even in the transient, i.e., we impose that at every iteration the polynomial defined by (42) satisfies $\partial(Q_i) \leq n$. This is achieved by imposing (39) and (43) at each adaptation step. Note that the "corrections" performed by Algorithm 5.1, namely, step (b1) and step (c2) are defined in a quite reasonable way. In fact, the substitution of R_{i-1} and P_{i-1} with the solution of (41) is expected "not to perturb too much" R_i, P_i, at least in the proximity of an equilibrium point. In the same way, in step (c2), we search for polynomial variations $\Delta P_i', \Delta R_i'$, which are "as close as possible" to $\Delta P_i, \Delta R_i$, but satisfy (46).

The introduction of the considered modifications in the basic algorithm does not cause a great increase in the computational complexity. Namely, the main additional complications are the solution of the optimization problem (45),(46), which is a quadratic problem with only equality (linear) constraints (the solution of this problem can be trivially found analitically), and the solution of the Bezoutian equation (41). Actually, (b2) remains in general the most complicated step in the algorithm.

Due to the modifications introduced in Algorithm 5.1, with respect to Algorithm 4.1, we can now give the following result.

Theorem 5.2. *Refer to Algorithm 5.1 (either for infinite horizon or m-step quadratic optimization) applied to an unknown system having structure (1). Then, assume that the identification-adaptation algorithm has reached an equilibrium point $(\mathcal{A}, \mathcal{B}, \mathcal{D}, R, P, T)$ characterized by the following facts:*

(i) the joint input-output process (u_t, y_t) is stationary;

(ii) $\gcd(R, P) = 1$;

(iii) $\partial(R, P) = n$;

(iv) $\mathcal{A}R - \mathcal{B}P$ *is asymptotically stable;*

Moreover assume that the following properties hold for the true system

(v) $\gcd(A, B) = 1$;

(vi) $\partial(A, B) = n$;

(vii) the same as (i) in Theorem 4.2;

(viii) the same as (ii) in Theorem 4.2;

Then the equilibrium point is whitening, (R, P, T) *coincides with the optimal (minimal order) regulator* (R°, P°, C) *for the true system, and* $(\mathcal{A}, \mathcal{B}, \mathcal{D})$ *is an implicit model associated with this regulator.*

□

Proof. It can be immediately seen that the same result of Theorem 5.1 can be also stated with reference to Algorithm 5.1 (instead of Algorithm 4.1). Then, note that assumption (i) assures that the convergence yields a stabilizing regulator for the true system model. Then, it is immediate to see that the fulfilment of the assumptions of the present Theorem (together with the structure of Algorithm 5.1) implies the fulfilment of all assumptions of Theorem 5.1, thus proving the thesis.

■

Note that assumptions i) $\div$ iv) in Theorem 5.2 refer to the parameters characterizing the reached equilibrium point, whereas assumptions v) $\div$ viii) refer to the unknown true system. The latter assumptions can be considered as generally verified (for the considered control criteria). Thus we can summarize, roughly speaking, the result provided by Theorem 5.2, in the following way: "the only possible equilibrium point (with the meaning stated in Definition 4.1) of Algorithm 5.2 is that yielding the optimal (minimum order) controller". This means that Algorithm 5.1 enjoys what is generally called "self-tuning property". The introduction of the modifications in Algorithm 5.1 (with respect to Algorithm 4.1) is essential in this respect, since it implies that condition (d) in Theorem 5.1 is always satisfied.

VI. SIMULATION RESULTS

An interactive menu-oriented software package has been developed [20], which allows to design and simulate various classes of stochastic adaptive control schemes for ARMAX models quadratic optimal control criteria. Several robust numerical techniques have been used as regards the implementation of the recursive identification module. Actually, a "factorized version" of the least squares algorithm [21] has been used. Besides, in such a module, there is the possibility of using the Improved Least Squares technique [22] (which automatically sets and adjusts the forgetting factor).

In the following we briefly report some simulation results obtained by using Algorithm 5.1.

An ARMAX system, characterized by the following polynomials, has been considered:

$$A(q^{-1}) = 1 - 0.7\,q^{-1}$$

$$B(q^{-1}) = 0.2\,q^{-1}$$

$$C(q^{-1}) = 1 + 0.3\,q^{-1}$$

with a system noise variance set to 0.1. An infinite–horizon quadratic cost was considered characterized by a control weight equal to 0.1. The optimal regulator which could be synthesized on the basis of the exact knowledge on the true system model is

$$R^{\circ}(q^{-1}) = 1 + 0.195\,q^{-1}$$

$$P^{\circ}(q^{-1}) = -1.835$$

$$T^{\circ}(q^{-1}) \equiv C(q^{-1}) = 1 + 0.3\,q^{-1}$$

The order of the model to be identified was set to 2. A single coefficient was fixed, namely $\delta_1 = 0$, according to the specification in Theorem 4.2 and the ILS identification technique was used. The implicit model associated with the optimal regulator is

$$\mathcal{A}(q^{-1}) = 1 - q^{-1}$$

$$\mathcal{B}(q^{-1}) = 0.2\,q^{-1} + 0.103\,q^{-2}$$

$$\mathcal{D}(q^{-1}) = -0.163\,q^{-2}$$

The dither noise variance was set to 0.01 and 3000 simulation/identification/adaptation steps were performed, which were sufficient for a good convergence. Fig. 1 shows the time–behaviours of the parameters of the implicit model and of the regulator. 5 minutes were sufficient on a PC486, to perform the 3000 simulation steps.

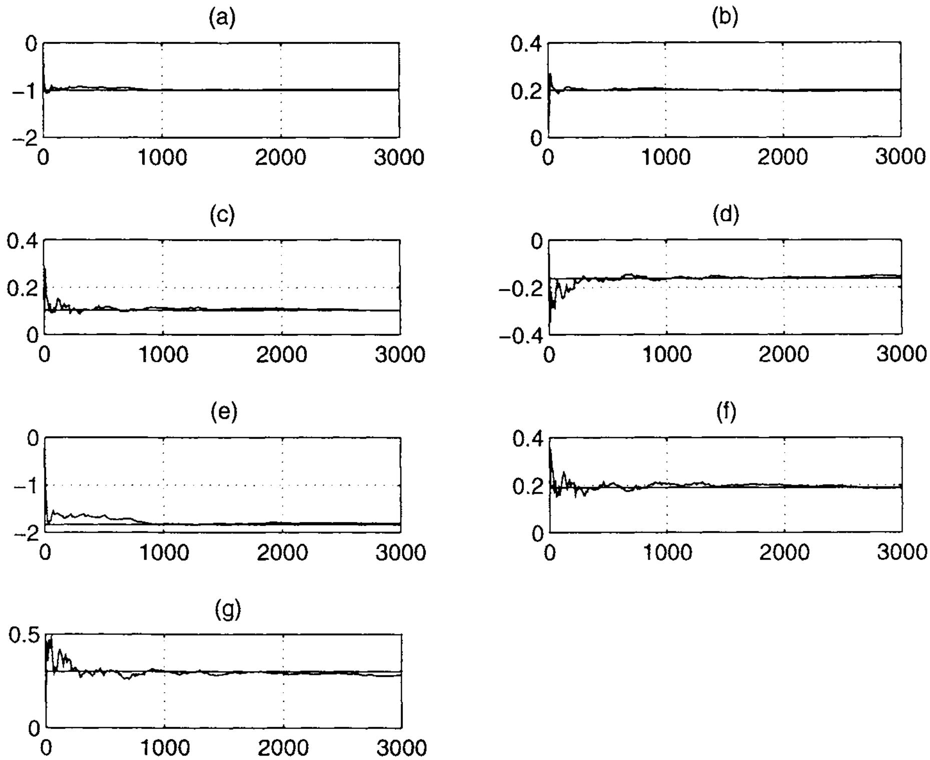


Fig. 1. Time behaviours of the identified model parameters ((a): α_1; (b): β_1; (c): β_2; (d): δ_2) and of the adaptive regulator coefficients ((e): p_0; (f): r_1; (g): t_1). The straight lines denote the corresponding optimal values.

VII. CONCLUSIONS

This work has presented a general theory which can be viewed as a basis for the proper use of ARX implicit models in adaptive control schemes. Necessary and sufficient conditions for the existence of such models have been given. Moreover, the possible use of such models has been discussed in order to design adaptive algorithms based on finite and infinite horizon quadratic criteria.

A particular way of exploiting the information provided by an implicit model has been considered. The main feature of this approach is that of obtaining, at every iteration of the algorithm, the new regulator not on the basis of the identified prediction model only, but also making use of the equation of the previously acting regulator. This is accomplished by solving a "variational control problem" and allows to develop adaptive control algorithms based on either finite or infinite horizon quadratic optimization.

Some theoretical results have been reported showing that the developed algorithm admits the optimal control law (for the true system model) as an equilibrium point, and that a proper structuring of such algorithm assures that it admits a unique whitening equilibrium point coinciding with the optimal regulator. Furthermore, a technical way to test whether a generic convergence point coincides with this unique whitening equilibrium point has been indicated. On the basis of such a result, a modification to the basic algorithm has been described which ensures the enjoyment the "self-tuning property" (that is, the algorithm, whenever converges, yields the optimal controller). Some simulation results show the effectiveness of the proposed approach.

VIII. REFERENCES

1 K. J. Åström and B. Wittenmark, "On Self Tuning Regulators," *Automatica* **9**, pp. 185-199 (1973).

2 D. W. Clarke and P. J. Gawthrop, "Self-Tuning Controller," *Proc. IEE* **122D**, pp. 929-934 (1975).

3 V. Borisson, "Self-tuning regulators for a class of multivariable systems," *Automatica* **15**, pp. 209-215 (1979).

4 H. N. Koivo, "A multivariable self-tuning controller," *Automatica* **16**, pp. 351-366 (1980).

5 L. Keviczky and K. S. P. Kumar, "Multivariable self-tuning regulator with generalized cost function, *Int. J. of Control* **33**, pp. 913-921 (1981).

6 M. M. Bayoumi, K. Y. Wong, and M. A. El-Bagoury, "A self-tuning regulator for multivariable systems," *Automatica* **17**, pp. 575-592 (1981).

7 P. E. Wellstead, D. Prager, and P. Zanker, "Pole assignment self-tuning regulator," *Proceedings of the IEE* **126**, pp. 781-787 (1979).

8 A. T. Allidina and F. M. Hughes, "Generalized self-tuning controller with pole-assignment," *Proc. IEE* **127D**, pp. 13-18 (1980).

9 V. Peterka, "Predictor-based self-tuning control," *Automatica* **20**, pp. 39-50 (1984).

10 M. J. Grimble, "Implicit and explicit LQG self-tuning controllers," *Automatica* **20**, pp. 661-669 (1984).

11 E. Mosca, *Optimal, predictive, and adaptive control*, Prentice Hall, NJ (1995).

12 G. Casalino, F. Davoli, R. Minciardi and G. Zappa, "On Implicit Modelling Theory: Basic Concepts and Applications to Adaptive Control," *Automatica* **23**, pp. 189-201 (1987).

13 G. Casalino, F. Davoli, and R. Minciardi, "Finite and infinite horizon LQ adaptive controllers based on implicit model identification," *Proc. 24th IEEE Conf. on Decision and Control*, Fort Lauderdale, pp. 853-855 (1985).

14 G. Casalino, F. Davoli, and R. Minciardi, "Extended Implicit Models and Application to LQ Adaptive Optimization," *Proc. 2° IFAC Workshop on Adaptive Systems in Control and Signal Processing*, Lund, Sweden, pp. 73-79 (1986).

15 E. Mosca and G. Zappa, "ARX modelling of controlled ARMAX plants and LQ adaptive controllers," *IEEE Trans. on Autom. Control* **AC-34**, pp. 371-375 (1989).

16 E. Mosca and J. M. Lemos, "A semi-infinite horizon LQ self-tuning regulator for ARMAX plants based on RLS," *Automatica*, **28**, pp. 401-406 (1992).

17 G. Casalino and R. Minciardi, "Further results on implicit models with application to LQ adaptive optimization," *Automatica* **28**, No. 2, pp.407-410 (1992).

18 G. Casalino, R. Minciardi, and T. Parisini, "Development of a new self-tuning control algorithm for finite and infinite horizon quadratic adaptive optimization", *Int. J. of Adaptive Control and Signal Processing* **5**, pp. 405-425 (1991).

19 P. Dorato and A. H. Levis, "Optimal linear regulators: the discrete–time case," *IEEE Trans. on Autom. Control* **AC-16**, pp. 613–620 (1971).

20 G. Casalino, R. Minciardi, C. Morelli, and T. Parisini, "ADPAC: A design and simulation tool for stochastic adaptive controllers," in *Recent Advances in Computer–Aided Control Systems Engineering*, M. Jamshidi and C. J. Herget (Eds.), Elsevier (1992).

21 G.J. Bierman, *Factorization methods for Discrete Sequential Estimation*, Academic Press (1977).

22 N. R. Sripada and D. G. Fisher, "Improved Least Squares Estimation," *Int. J. of Control* **46**, pp. 1889-1913 (1987).

APPENDIX.

A. PROOF OF THEOREM 2.1

(*Only if*). Assume the existence of an extended implicit model characterized by the triple $(\mathcal{A}, \mathcal{B}, \mathcal{D})$. Then, by equating the closed–loop transfer functions between the input e_i and the output y_i corresponding to the true system model and to the implicit model, respectively, one has

$$\frac{RC}{AR - BP} = \frac{R}{\mathcal{A}R - \mathcal{B}P} \tag{A1}$$

This, in turn, requires

$$AR - BP = QC \tag{A2}$$
$$\mathcal{A}R - \mathcal{B}P = Q \tag{A3}$$

which proves the necessity of condition $i)$, $ii)$. Furtherly, by equating the closed–loop transfer functions between the input η_i and the output y_i corresponding to the true system model and to the implicit model, respectively, one has

$$\frac{TB}{AR - BP} = \frac{T\mathcal{B} + \mathcal{D}R}{\mathcal{A}R - \mathcal{B}P} \tag{A4}$$

Then, since we have already proved that (7) and (10) hold, we have

$$AR - BP = (\mathcal{A}R - \mathcal{B}P)C \tag{A5}$$

and thus (A4) can be written as

$$\frac{TB}{C} = T\mathcal{B} + \mathcal{D}R \tag{A6}$$

which, the r.h.s. being a polynomial, implies (9), where $S(q^{-1})$ must be monic, since $T(q^{-1})$ and $C^{\circ}(q^{-1})$ are monic. Then, using (9), one has

$$SB^{\circ} = SC^{\circ}\mathcal{B} + \mathcal{D}R \tag{A7}$$

which coincides with (14) and implies (11).

Let us now prove that condition *iii*) can be substituted by condition *iii bis*). First, let us prove that *iii bis*) is implied by *i*), *ii*), *iii*) and *iv*). To this end, let us factorize polynomial $C(q^{-1})$ as follows

$$C = \varphi\,[\mathrm{gcd}\,(A,B,C)]\,\frac{\mathrm{gcd}\,(B,C)}{\mathrm{gcd}\,(A,B,C)}\,\frac{\mathrm{gcd}\,(A,C)}{\mathrm{gcd}\,(A,B,C)} \tag{A8}$$

that is $C(q^{-1}) = \varphi\,\psi\,\gamma\,\delta$, with implicit symbol definition. Then, clearly $C^{\circ}(q^{-1}) = \varphi\delta$, $\bar{C}^{\circ}(q^{-1}) = \varphi\gamma$. The implication we want to ensure is proved if we can prove that $S(q^{-1})$ is a multiple of γ (since γ cannot be a divisor of $C^{\circ}(q^{-1})$ by definition). In order to prove this, let us use (11), which states that

$$\frac{S(B^{\circ} - C^{\circ}\mathcal{B})}{R}$$

is a polynomial (for some pair $(\mathcal{A},\ \mathcal{B})$ satisfying (10)). But

$$\frac{B - C\mathcal{B}}{R} = \frac{A - C\mathcal{A}}{P} \tag{A9}$$

due to (A5), and thus even

$$\frac{S(A - C\mathcal{A})}{P\,\mathrm{gcd}\,(B,C)} \tag{A10}$$

must be a polynomial. Relationship (A10) can be rewritten as

$$\frac{S[A/\mathrm{gcd}\,(B,C)] - C^{\circ}\mathcal{A}}{P} = \frac{S(A/\psi\,\gamma) - C^{\circ}\mathcal{A})}{P} \tag{A11}$$

Then, keeping in mind that (A11) is a polynomial and that $A(q^{-1})$ cannot be a multiple of γ, it clearly appears that γ must be a divisor of $S(q^{-1})$. Note that , in a quite symmetrical way, one could prove that *i*), *ii*), *iii bis*), *iv*) imply *iii*).

Finally, let us prove that condition *iv*) can be substituted by condition *iv bis*). To this end, consider a triple $(\mathcal{A},\mathcal{B},\mathcal{D})$ for which (10), (11) and (14) are satisfied. Then, rewrite (14) as

$$\mathcal{D} = \frac{T}{C^{\circ}}\,\frac{B^{\circ} - C^{\circ}\mathcal{B}}{R} = \frac{T(B - C\mathcal{B})}{RC} \tag{A12}$$

Then, using again (A9), we have

$$\mathcal{D} = \frac{T(A - C\mathcal{A})}{PC} = \frac{T}{\bar{C}^{\circ}} \frac{A^{\circ} - \bar{C}^{\circ}\mathcal{A}}{P} \tag{A13}$$

which coincides with (15). This in turn proves that condition *iv bis*) holds. Moreover, note again that, in a quite symmetric way, one could prove that *i*), *ii*), *iii*), *iv bis*) imply *iv*).

(*If*). Assume now that conditions *i*), *ii*), *iii*), *iv*) hold, and recall the assumption of closed-loop asymptotic stability. This assumption implies that the behaviour of the output y_i is fully characterized by the following transfer functions

$$\frac{RC}{AR - BP}; \quad \frac{TB}{AR - BP} \tag{A14}$$

from the input e_i and the input η_i, respectively. But, by assumption, (7) holds and at least a triple $(\mathcal{A}, \mathcal{B}, \mathcal{D})$ exists satisfying (10), (11) and (14). Then, let us consider the system

$$\mathcal{A}(q^{-1})\bar{y}_i = \mathcal{B}(q^{-1})\bar{u}_i + \mathcal{D}(q^{-1})\eta_i + e_i \tag{A15}$$

controlled by

$$R(q^{-1})\bar{u}_i = P(q^{-1})\bar{y}_i + T(q^{-1})\eta_i \tag{A16}$$

Of course, even (A15), (A16) represent an asymptotically stable closed-loop system (since the closed-loop poles are the zeros of $\mathcal{A}R - \mathcal{B}P = (AR - BP)/C = Q$). Then, in order to characterize the behaviour of the output $\bar{y}_i$, it is sufficient to consider the transfer functions

$$\frac{R}{\mathcal{A}R - \mathcal{B}P}; \quad \frac{T\mathcal{B} + \mathcal{D}R}{\mathcal{A}R - \mathcal{B}P} \tag{A17}$$

from the input e_i and the input η_i, respectively. Then, (7), (10), (11), (14) imply the pairwise equality of the transfer functions in (A14) and (A17). Thus, y_i is asymptotically equal to $\bar{y}_i$ for any couple of realizations of the sequences $\{e_i\}$ and $\{\eta_i\}$, which actually proves that the considered triple $(\mathcal{A}, \mathcal{B}, \mathcal{D})$ represents an implicit model for the considered system. Finally, the possibility of substituting *iii*) with *iii bis*) and *iv*) with *iv bis*) is demonstrated by the considerations contained in the "only if" part.

■

B. PROOF OF THEOREM 2.2

To begin with, let us prove that condition *iii ter*) implies that the two expressions for the term $\mathcal{D}(q^{-1})$ in (14)–(15) actually represent a polynomial if $[\mathcal{A}(q^{-1}), \mathcal{B}(q^{-1})]$ satisfy the Bezoutian equation (10), and are equivalent. Actually from condition *iii ter*), we have

$$T = ZC_\circ \mathrm{gcd}\,(R, P) \tag{A18}$$

$Z(q^{-1})$ being a suitable polynomial. Moreover, let us factorize polynomial $C(q^{-1})$ into

$$C = \varphi\,[\mathrm{gcd}\,(A, B, C)]\,\frac{\mathrm{gcd}\,(B, C)}{\mathrm{gcd}\,(A, B, C)}\;\frac{\mathrm{gcd}\,(A, C)}{\mathrm{gcd}\,(A, B, C)} \tag{A19}$$

which we can rewrite as $C(q^{-1}) = \varphi\,\psi\,\gamma\,\delta$, with all symbols defined by their positions in the sequence. Then, referring to the polynomials considered in connection with (14)–(15), one obtains

$$C_\circ = \varphi\gamma\delta = C^\circ\gamma \tag{A20}$$

$$S = Z\,\gamma\,\mathrm{gcd}\,(R, P) \tag{A21}$$

Now, let us consider a pair $(\mathcal{A}, \mathcal{B})$ satisfying the Bezoutian equation (10). Then, this pair also satisfies

$$(\mathcal{A}R - \mathcal{B}P)C = AR - BP \tag{A22}$$

which can be rewritten as

$$(A - \mathcal{A}C)R = (B - \mathcal{B}C)P \tag{A23}$$

or even as

$$(A - \mathcal{A}C)R_\circ = (B - \mathcal{B}C)P_\circ \tag{A24}$$

where $(R_\circ, P_\circ) \triangleq (R, P)/\mathrm{gcd}\,(R, P)$. (A24) implies the existence of a polynomial $F(q^{-1})$ such that

$$A - \mathcal{A}C = P_\circ F\,; \quad B - \mathcal{B}C = R_\circ F \tag{A25}$$

which, in turn, implies

$$(A_\circ - \mathcal{A}C_\circ)\psi = P_\circ F\,; \quad (B_\circ - \mathcal{B}C_\circ)\psi = R_\circ F \tag{A26}$$

where

$$(A_\circ, B_\circ, C_\circ) \triangleq (A, B, C)/\psi$$

with $\psi \triangleq \mathrm{gcd}\,(A, B, C)$. Hence, ψ must be a factor of $F(q^{-1})$. Then, define $F_\circ(q^{-1}) \triangleq F(q^{-1})/\psi$ and rewrite

$$(A_\circ - \mathcal{A}C_\circ) = P_\circ F_\circ\,; \quad (B_\circ - \mathcal{B}C_\circ) = R_\circ F_\circ \tag{A27}$$

In particular, from the second expression in (A27), we have

$$(B^\circ - \mathcal{B}C^\circ)\gamma = R_\circ F_\circ \tag{A28}$$

Finally, (A21) and (A28) yield

$$S(B^\circ - \mathcal{B}C^\circ) = Z\,\gamma\,\mathrm{gcd}\,(R,P)\,(B^\circ - \mathcal{B}C^\circ) = ZRF_\circ \tag{A29}$$

which proves that the expression defining $\mathcal{D}(q^{-1})$ in (14) actually represents a polynomial. The equivalence of the two ewpressions (14) and (15) can be easily proved by means of (A23).

Now, let us prove the main implication. First, let us observe that the assumption about closed–loop asymptotic stability implies that the behaviour of the output y_i is fully characterized by the following transfer functions

$$\frac{RC}{AR - BP};\quad \frac{TB}{AR - BP} \tag{A30}$$

from the input e_i and η_i, respectively.

Then, let us make use of the result in Theorem 2.1. which allows us to state, under assumptions *i*) and *ii*), that at least a pair $(\mathcal{A}, \mathcal{B})$ satisfying the Bezoutian equation (10) exists. Let us consider such a pair, and associate with $(\mathcal{A}, \mathcal{B})$ a third polynomial $\mathcal{D}$ as given by (14) or (15).

Now, we can build up a coupled system

$$\mathcal{A}(q^{-1})\bar{y}_i = \mathcal{B}(q^{-1})\bar{u}_i + \mathcal{D}(q^{-1})\eta_i + e_i \tag{A31}$$

$$R(q^{-1})\bar{u}_i = P(q^{-1})\bar{y}_i + T(q^{-1})\eta_i \tag{A32}$$

Clearly, even (A31) and (A32) represent an asymptotically stable closed–loop system (as the closed–loop poles are the zeros of $\mathcal{A}R - \mathcal{B}P = (AR - BP)/C = Q)$. Then, in order to characterize the behaviour of the output y_i, it is sufficient to consider the transfer functions

$$\frac{R}{\mathcal{A}R - \mathcal{B}P};\quad \frac{T\mathcal{B} + \mathcal{D}R}{\mathcal{A}R - \mathcal{B}P} \tag{A33}$$

from the input e_i and η_i, respectively. But, from (7), (10) and (14)–(15), it is immediate to deduce the pairwise equality of the transfer functions in (A30) and (A33). Thus, y_i is asymptotically equal to $\bar{y}_i$ for any couple of realizations of the sequences $\{e_i\}$ and $\{\eta_i\}$. This actually proves that the triple $(\mathcal{A}, \mathcal{B}, \mathcal{D})$ represents an implicit model of the considered system.

■

C. PROOF OF THEOREM 5.1

The proof follows a reasoning line similar to that in an analogous proof given in [7].

Before entering into the details of the proof, let us mention three basic implications of the assumptions in the Theorem.

First of all, the assumption that the algorithm has reached an equilibrium point characterized by

$$\mathcal{A}(q^{-1})\, y_i = \mathcal{B}(q^{-1})\, u_i + \mathcal{D}(q^{-1})\, \eta_i + \xi_i \tag{A34}$$

(where $\{\xi_i\}$ is the prediction error sequence), the assumption about the order of the extended implicit model $(\mathcal{A}, \mathcal{B}, \mathcal{D})$, and standard Least Squares properties together imply that:

$$S_{\xi y}(\tau) \triangleq E(\xi_i\, y_{i-\tau}) = 0 \quad \tau = 1, 2, \ldots, n+1 \tag{A35a}$$

$$S_{\xi u}(\tau) \triangleq E(\xi_i\, u_{i-\tau}) = 0 \quad \tau = 1, 2, \ldots, n+1 \tag{A35b}$$

$$S_{\xi \eta}(\tau) \triangleq E(\xi_i\, \eta_{i-\tau}) = 0 \quad \tau = 2, \ldots, n+1 \tag{A35c}$$

The third equation does not hold for $\tau = 1$, since we assume here that δ_1 is the coefficient which is arbitrarly fixed in the extended implicit model to be identified. Of course, this choice does not affect the generality of the following discussion.

Moreover, assumption (a) can be expressed by a matricial condition, that is

$$(a) \Longleftrightarrow rank\,\{V_1\} = 2n \tag{A36}$$

where V_1 is the resultant matrix of (A, B) given by

$$V_1 = \begin{bmatrix} 1 & a_1 & a_2 & \ldots & a_n & 0 & 0 & \ldots & 0 \\ 0 & 1 & a_1 & a_2 & \ldots & a_n & 0 & \ldots & 0 \\ \vdots & & & \ddots & \ddots & & \ddots & & \\ 0 & \ldots & \ldots & 1 & a_1 & a_2 & \ldots & \ldots & a_n \\ 0 & b_1 & b_2 & \ldots & b_n & 0 & 0 & \ldots & 0 \\ 0 & 0 & b_1 & \ldots & & b_n & 0 & \ldots & 0 \\ 0 & 0 & \ddots & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & & \ddots & 0 \\ 0 & \ldots & \ldots & 0 & 0 & b_1 & b_2 & \ldots & b_n \end{bmatrix} \tag{A37}$$

and, in the same way, as regards the assumption (c),

$$(c) \Longleftrightarrow rank\,\{V_2\} = 2n \tag{A38}$$

where

$$V_2 = \begin{bmatrix} 1 & r_1 & r_2 & \dots & r_n & 0 & 0 & \dots & 0 \\ 0 & 1 & r_1 & \dots & & r_n & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ 0 & \dots & \dots & 1 & r_1 & r_2 & \dots & \dots & r_n \\ p_0 & p_1 & p_2 & \dots & p_n & 0 & 0 & \dots & 0 \\ 0 & p_0 & p_1 & \dots & & p_n & 0 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & & \ddots & 0 \\ 0 & \dots & \dots & 0 & p_0 & p_1 & p_2 & \dots & p_n \end{bmatrix} \tag{A39}$$

At this point, let us enter into the main body of the proof, which will be subdivided into four steps for the sake of convenience.

Step 1 : Decomposition

The prediction error sequence is given by (A34), which can be rewritten as

$$\xi_i = \mathcal{A}(q^{-1})\, y_i - \mathcal{B}(q^{-1})\, u_i - \mathcal{D}(q^{-1})\, \eta_i \tag{A40}$$

where y_i and u_i represent the outputs of the closed-loop system (1),(3), driven by the couple of independent external signals e_i , η_i , which are both white stationary noises. Then, by using superposition, we can write :

$$\xi_i = \xi_i{}^e + \xi_i{}^\eta\,, \quad y_i = y_i{}^e + y_i{}^\eta\,, \quad u_i = u_i{}^e + u_i{}^\eta \tag{A41}$$

where the superscripts e, η denote the contributions to the global signals due to e_i and η_i, respectively. Obviously, because of the assumed independence of e_i and η_i , each term superscribed by e is independent by any term superscribed by η , and vice versa. The above fact implies that equations (A35) can be expressed equivalently by

$$S_{\xi^e y^e}(\tau) = 0 \quad \tau = 1, 2, \dots, n+1 \tag{A42a}$$

$$S_{\xi^e u^e}(\tau) = 0 \quad \tau = 1, 2, \dots, n+1 \tag{A42b}$$

$$S_{\xi^\eta y^\eta}(\tau) = 0 \quad \tau = 1, 2, \dots, n+1 \tag{A42c}$$

$$S_{\xi^\eta u^\eta}(\tau) = 0 \quad \tau = 1, 2, \dots, n+1 \tag{A42d}$$

$$S_{\xi^\eta \eta}(\tau) = 0 \quad \tau = 2, \ldots, n+1 \tag{A42e}$$

Step 2 : Analysis of the contribution due to e_i

In the following analysis $\eta_i \equiv 0$ will be assumed.

Let us consider $Q(q^{-1}) = \mathcal{A}(q^{-1})\,R(q^{-1}) - \mathcal{B}(q^{-1})\,P(q^{-1})$, characterized by $\partial(Q) \leq n$ (assumption (d)), and asymptotically stable (assumption (b)). Then, from (3) and (A40) it follows that

$$Q(q^{-1})\,y_i{}^e = R(q^{-1})\,\xi_i{}^e \tag{A43a}$$

$$Q(q^{-1})\,u_i{}^e = P(q^{-1})\,\xi_i{}^e \tag{A43b}$$

or even

$$y_i{}^e = R(q^{-1})\,w_i \tag{A44a}$$

$$u_i{}^e = P(q^{-1})\,w_i \tag{A44b}$$

where w_i is defined by $Q(q^{-1})\,w_i = \xi_i{}^e$ and turns out to be a stationary stochastic sequence, due to the asymptotic stability of Q.

Taking into account the definition of the matrix V_2, given by (A39), and the relations (A44), the first n of (A42a) and the first n of (A42b) can be rewritten as

$$V_2\;[S_{\xi^e w}(1), \ldots, S_{\xi^e w}(2n)]^T = 0 \tag{A45}$$

which implies, due to the assumption (b) and to the equivalence (A38),

$$S_{\xi^e w}(\tau) = 0 \quad \tau = 1, 2, \ldots, 2n \tag{A46}$$

Then, by means of the model equation (A34) and of the system and regulator equations (1),(3), it is possible to obtain a relation between $\{\xi_i{}^e\}$ and $\{e_i\}$, that is,

$$(A\,R - B\,P)\;\xi_i{}^e = Q\,C\,e_i \tag{A47}$$

where obviously

$$\partial(AR - BP) \leq 2n \tag{A48}$$

and, due to assumption (d),

$$\partial(QC) \leq 2n \tag{A49}$$

At this point, let us multiply both sides of (A47) by $w_{i-(2n+1)}$ and take expectation; we obtain

$$\begin{aligned} &S_{\xi^e w}(2n+1) + l_1 S_{\xi^e w}(2n) + \ldots + l_{2n} S_{\xi^e w}(1) \\ &\qquad = S_{ew}(2n+1) + g_1 S_{ew}(2n) + \ldots + g_{2n} S_{ew}(1) \end{aligned} \tag{A50}$$

where $l_1, \ldots, l_{2n}$ are the coefficients of the monic polynomial

$$A\,R - B\,P \tag{A51}$$

$g_1, \ldots, g_{2n}$ are the ones of the monic polynomial $Q(q^{-1})C(q^{-1})$. The l.h.s. of the (A50) reduces to $S_{\xi^e w}(2n+1)$, due to (A46) whereas the r.h.s. is zero, since the future values of e_i are independent of the past values of w_i. Thus (A50) reduces to $S_{\xi^e w}(2n+1) = 0$. The same considerations can be repeated also for $S_{\xi^e w}(2n+2)$, and so on. Then, the relation (A46) and the above discussion imply that

$$S_{\xi^e w}(\tau) = 0 \quad \tau \geq 1 \tag{A52}$$

Now observe that, on the basis of the definition of w_i, it follows that

$$S_{\xi^e \xi^e}(\tau) = S_{\xi^e w}(\tau) + q_1 S_{\xi^e w}(\tau+1) + \ldots + q_n S_{\xi^e w}(\tau+n) \tag{A53}$$

where $q_1, \ldots, q_n$ are the coefficients of the monic polynomial $Q(q^{-1})$. Then, from (A53) and (A52), we obtain

$$S_{\xi^e \xi^e}(\tau) = 0 \quad \tau \geq 1 \tag{A54}$$

which shows that $\{\xi_i{}^e\}$ is a white sequence.

But, on the basis of (A47), $\xi_i{}^e$ can be represented by a linear combination of $e_i, e_{i-1}, \ldots,$. Thus, it follows that

$$\xi_i{}^e = e_i \tag{A55}$$

which implies, due again to (A47),

$$A\,R - B\,P = Q\,C = (A\,R - B\,P)\,C \tag{A56}$$

Step 3 : Analysis of the contribution due to $\{\eta_i\}$

In the following discussion $e_i \equiv 0$ will be assumed. Then, taking into account the system and regulator equations (1),(3) we can write

$$\left[A(q^{-1})\,R(q^{-1}) - B(q^{-1})\,P(q^{-1})\right] y_i{}^\eta = B(q^{-1})\,T(q^{-1})\,\eta_i \tag{A57a}$$

$$\left[A(q^{-1})\,R(q^{-1}) - B(q^{-1})\,P(q^{-1})\right] u_i{}^\eta = A(q^{-1})\,T(q^{-1})\,\eta_i \tag{A57b}$$

or, due to (A56),

$$Q(q^{-1})\,C(q^{-1})\,y_i{}^\eta = B(q^{-1})\,T(q^{-1})\,\eta_i \tag{A58a}$$

$$Q(q^{-1})\, C(q^{-1})\, u_i{}^{\eta} = A(q^{-1})\, T(q^{-1})\, \eta_i \tag{A58b}$$

Now we can express these relations in the following way:

$$y_i{}^{\eta} = B(q^{-1})\, v_i \tag{A59a}$$

$$u_i{}^{\eta} = A(q^{-1})\, v_i \tag{A59b}$$

where v_i is defined by

$$QCv_i = T\eta_i \tag{A60}$$

It is important to point out that v_i is a stochastic stationary sequence, due to the asymptotic stability of Q (assumption (b)).

Now by taking into account the definition of matrix V_1, given by (A37), and the relations (59), the first n relations of (42c) and the first n relations of (42d), can be rewritten as

$$V_1 \left[S_{\xi^{\eta}v}(1), \ldots, S_{\xi^{\eta}v}(2n)\right]^T = 0 \tag{A61a}$$

In a similar way the last n of (42c) and the last n of (42d), can be rewritten as

$$V_1 \left[S_{\xi^{\eta}v}(2), \ldots, S_{\xi^{\eta}v}(2n+1)\right]^T = 0 \tag{A61b}$$

Since the matrix V_1 is non singular, due to assumption (a) and to equivalence (A36), relations (61) imply

$$S_{\xi^{\eta}v}(\tau) = 0 \quad \tau = 1, \ldots, 2n+1 \tag{A62}$$

At this point, let us analyze the relation between $\{\xi_i{}^{\eta}\}$ and $\{\eta_i\}$. In order to determine such relation, let us substitute (57) into (A40), thus obtaining

$$\begin{aligned}&\left[A(q^{-1})\, R(q^{-1}) - B(q^{-1})\, P(q^{-1})\right]\, \xi_i{}^{\eta} \\ &= \Big\{ T(q^{-1}) \left[\mathcal{A}(q^{-1})\, B(q^{-1}) - A(q^{-1})\, \mathcal{B}(q^{-1})\right] \\ &-\mathcal{D}(q^{-1}) \, \left[A(q^{-1})\, R(q^{-1}) - B(q^{-1})\, P(q^{-1})\right] \Big\}\, \eta_i\end{aligned} \tag{A63}$$

Then, consider the identity (whose proof is immediate on the basis of (A56))

$$(\mathcal{A}\, B - A\, \mathcal{B}) = \frac{(B - \mathcal{B}C)\, (\mathcal{A}\, R - \mathcal{B}\, P)}{R} \tag{A64}$$

which can be rewritten as

$$\mathcal{A}B - A\mathcal{B} = SQ \tag{A65}$$

being

$$S = \frac{[B - \mathcal{B}\,C]}{R} = \frac{[A - \mathcal{A}\,C]}{P} \tag{A66}$$

necessarily a polynomial. Thus due to (A65) and (A56), equation (A64) can be rewritten as

$$Q(q^{-1})\,C(q^{-1})\,\xi_i{}^{\eta} = \left[T(q^{-1})\,S(q^{-1})\,Q(q^{-1}) - \mathcal{D}(q^{-1})\,Q(q^{-1})\,C(q^{-1})\right]\,\eta_i \tag{A67}$$

or even, taking into account the asymptotic stability of $Q(q^{-1})$,

$$C(q^{-1})\,\xi_i{}^{\eta} = \left[T(q^{-1})\,S(q^{-1}) - \mathcal{D}(q^{-1})\,C(q^{-1})\right]\,\eta_i \tag{A68}$$

It has to be remarked that $\partial(S) \leq n + 1$. Now, multiplying both sides of (A68) by $v_{i-(2n+2)}$ and taking expectation, we obtain

$$\begin{aligned} &S_{\xi^\eta v}(2n+2) + c_1 S_{\xi^\eta v}(2n+1) + \ldots + c_n S_{\xi^\eta v}(n+2) \\ &\quad = f_1 S_{\eta v}(2n+1) + \ldots + f_{2n+1} S_{\eta v}(1) \end{aligned} \tag{A69}$$

where $f_1, \ldots, f_{2n+1}$ are the coefficients of the polynomial $(T\,S - \mathcal{D}\,C)$, which has the zero-th order coefficient equal to zero, as happens for $S(q^{-1})$ and $\mathcal{D}(q^{-1})$.

The l.h.s. of (A69) is then equal to $S_{\xi^\eta v}(2n+2)$, due to (A62); the r.h.s. is simply zero, since future values of η_i are independent from past values of v_i (see also (A60)).

The same argument can be carried out for $S_{\xi^\eta v}(2n+3)$, $S_{\xi^\eta v}(2n+4)$, and so on, in a similar way to that followed in the preceding step of the proof. Then, on the whole, we can conclude that

$$S_{\xi^\eta v}(\tau) = 0 \quad \tau \geq 1 \tag{A70}$$

But, from the relation (A60), we can state that

$$S_{\xi^\eta (QCv)}(\tau) = S_{\xi^\eta (T\eta)}(\tau) \quad \tau \geq 1 \tag{A71}$$

where the l.h.s. of this relation is zero due to (A70). Hence,in particular $S_{\xi^\eta (T\eta)}(1) = 0$. Then, it follows, taking in account (42e), that, on the whole,

$$S_{\xi^\eta \eta}(\tau) = 0 \quad \tau \geq 1 \tag{A72}$$

Moreover note that (A68) implies that $\xi_i{}^{\eta}$ is influenced by η_{i-1}, η_{i-2}, ...only.

Thus, (A72) leads to

$$\xi_i{}^{\eta} = 0 \quad \forall i \tag{A73}$$

From this relation, and taking again into account the (A68), it follows that

$$(ST - \mathcal{D}C) = 0 \tag{A74}$$

or even

$$\mathcal{D} = \frac{ST}{C} \tag{A75}$$

Let us observe now that the polynomials S and C are coprime, due to the definition of S (A66); in fact, if they were not, then the common factor of S and C, would also be a factor of A and B, which is excluded by the assumption (a) of the theorem.

Then, the following relations hold:

$$T = C \tag{A76a}$$

$$\mathcal{D} = S = \frac{(A - \mathcal{A}C)}{P} = \frac{(B - \mathcal{B}C)}{R} \tag{A76b}$$

Step 4 : Conclusion.

On the basis of superposition (A41), the sequence $\{\xi_i\}$ is a white one. This fact means that the relation (A34) represents an implicit model, which is moreover consistent with the fact that $(\mathcal{A}, \mathcal{B}, \mathcal{D})$ satisfy (76). As a consequence, the considered whitening equilibrium point is *the unique* whitening equilibrium point, and corresponds to the (minimal order) optimal control law.

This Page Intentionally Left Blank

INDEX

D

H

I

K

L

P

Q

R

S

T

U

V

W

ISBN 0-12-012779-2

90065

9 780120 127795