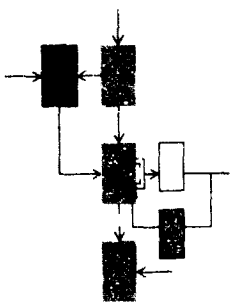
October, 1980

LIDS-TH-1055

Research Supported By:

ONR Grant N00014-77-C-0224
OSP No. 85008

NASA Grant NGL-22-009-124
OSP No. 76265



FAILURE DETECTION SYSTEM DESIGN METHODOLOGY

Edward Y. Chew

Laboratory for Information and Decision Systems

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

OCTOBER 1980 LIDS-TH-1055

FAILURE DETECTION SYSTEM DESIGN METHODOLOGY

by

Edward Y. Chow

This report is based on the unaltered thesis of Edward Y. Chow, submitted in partial fulfillment of the requirements for the degree of Doctor of Science at the Massachusetts Institute of Technology in October 1980. The research was conducted at M.I.T. Laboratory for Information and Decision Systems with support provided by the Office of Naval Research N00014-77-C-0224 and NASA/Ames Grant No. NGL-22-009-124.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

A FAILURE DETECTION SYSTEM DESIGN METHODOLOGY

by

EDWARD YIK CHOW

S.B., Massachusetts Institute of Technology
(1974)

S.M., Massachusetts Institute of Technology
(1976)

E.E., Massachusetts Institute of Technology
(1977)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

October, 1980

Signature of Author .....Edward Y Chow.....
Department of Electrical Engineering and Computer Science

Certified by .....Alan S. Willsky.....
Thesis Supervisor

Accepted by .....
Chairman, Departmental Committee on Graduate Students

ABSTRACT

The design of a failure detection and identification (FDI) system consists of designing a robust residual-generation process and a high-performance decision-making process. In this research the design of these two processes were examined separately.

Residual-generation is based on analytical redundancy. Redundancy relations that are insensitive to modelling errors and noise effects are important for designing robust residual-generation processes. The characterization of the concept of analytical redundancy in terms of a generalized parity space, as presented in this thesis, provided a framework in which a systematic approach to the determination of robust redundancy relations was developed.

The Bayesian approach was adopted for the design of high-performance decision processes. The FDI decision problem was formulated as a Bayes sequential decision problem. Since the optimal decision rule is incomputable, a methodology for designing suboptimal rules was proposed. A numerical algorithm was developed to facilitate the design and performance evaluation of suboptimal rules. This design approach was applied to an example, and the results were compared with those of Monte Carlo simulations.

ACKNOWLEDGEMENT

I wish to thank Prof. Alan Willisky, Prof. Wallace Vander Velde, Dr. Stanley Gershwin, and Prof. Michael Athans for their guidance during this thesis research. Their help and encouragement are deeply appreciated.

I would like to thank Bruce Walker and Joe Kimemia for many useful discussions concerning this thesis work. Also, I am especially indebted to my officemates, Peter Doerschuk and Jeff Lewis, who have helped in many ways during this research.

Finally, I would like to express my sincere appreciation to Fifa Monserrate for typing this thesis and Art Giordani for preparing the figures.

This research was supported in part by grants from the Office of Naval Research (N00014-77-C-0224) and NASA/AMES (NGL-22-009-124).

TABLE OF CONTENTS

LIST OF FIGURES

CONT. LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Physical systems are often subjected to unexpected changes, such as component failures and variations in operating conditions, that tend to degrade overall system performance. We will refer to such changes as failures, although there may not be any physical failure present. Maintaining a certain level of performance under failure is the objective of reliable system designs. In some cases, it is possible to design a system that is relatively insensitive to certain failures without explicitly detecting them. However, the inevitable tradeoff is reduced effectiveness of the system during normal conditions. Therefore, explicit failure detection and accommodation may be more desirable if such degraded overall performance must be avoided. Another situation where explicit failure detection and identification is required is one when an appropriate back-up actuator or sensor needs to be activated to replace the faulty one. Here, one needs to know which instrument should be used. Although failure detection and accommodation represent a single objective, it is often reasonable to assume that the appropriate remedy for each possible failure is known. From this perspective, the detection and identification of failures can be treated as a separate problem and this is the subject of this thesis research.

1.1 Problem Description

The study of failure detection and identification (FDI) in dynamical systems is based on the analysis of the structure and behavior of systems, which are described by mathematical models. In this research, we are

mainly concerned with the linear, time-invariant stochastic, discrete time model:

$$x(k+1) = Ax(k) + \sum_{j=1}^{q} b_j u_j(k) + \xi(k) \qquad (1\text{-}1)$$

$$y_j(k) = c_j x(k) + \eta_j(k), \qquad j=1,\ldots,m \qquad (1\text{-}2)$$

where x is the n-dimensional state vector, $u_1,\ldots,u_q$ are the q known actuator inputs, and $y_1,\ldots,y_m$ are the m sensor outputs (measurements); ξ and η are independent zero mean, white Gaussian (noise) sequences with covariance

$$E\{\xi(k)\xi'(t)\} = Q\delta_{k,t}$$

$$E\{\eta(k)\eta'(t)\} = R\delta_{k,t}$$

where $\delta_{k,t}$ is the Kronecker delta. The column vector b_j corresponds to the j-th actuator and input u_j, and the row vector c_j corresponds to the j-th sensor. Equations (1-1) and (1-2) are used to model a dynamical system in the normal mode, i.e. in the no-fail situation.

Failures represent abrupt changes. Hence, various failure modes (failure types) can be modelled as deviations from the normal mode. A faulty sensor may take the form of a change in c_j, a bias, or increased measurement noise in (1-2). A malfunctioning actuator may manifest itself as a shift in b_j, and an actuator "stuck" at a certain position that causes an input bias

may be described by a bias in (1-1). In some applications, the linear model (1-1)-(1-2) is used to represent the linearized behavior of a nonlinear system at a particular operating point. A change in the set point can result in a different set of system matrices, i.e. A, $\{b_j\}$, and $\{c_j\}$. Thus, shifts in all the system matrices are often necessary in order to model such a change.

Each failure is characterized by three attributes: 1) the failure mode or failure type (i), e.g. a biased sensor or a "stuck" actuator, 2) the failure time (τ)-the time at which the failure occurs, and 3) the magnitude (extent) of the failure (ν), e.g. the size of a sensor bias. By the very nature of a failure, these attributes are not known. Depending on the situation, not all three attributes are of equal importance. Consider, for instance, the problem of a failed sensor. With the availability of back-up sensors, being able to identify the failure mode (failed sensor) may provide acceptable overall performance. However, if we want to compensate an estimate of x based on, for example, the Kalman filter (KF) [1] for the error due to a failed sensor, we need to identify the failure time and the failure magnitude as well as the failure mode. When back-up sensors are not available, we have to make use of a degraded sensor. Then, we need to determine both the failure mode and the failure magnitude (e.g. the size of the bias that has developed in the sensor). However, it is sometimes necessary to estimate both τ and ν in order to do a good job of identifying i, even when the failure mode is the only important parameter. This is analogous to the problem of estimating a subset of the state variables of a system.

In order to obtain accurate estimates of these variables, it is sometimes necessary to use a full order filter to estimate the entire state vector. In any case, the detection of a failure (i,τ,ν) requires the examination of the measurement for the failure's characteristic effects.

In recent years, numerous approaches (e.g. the voting scheme [2][3], the generalized likelihood ratio (GLR) method [4][5], the multiple model method [5][6] and the detection filters of Beard [7] and Jones [8]) have been developed to perform FDI in dynamical systems with linear stochastic models. A comprehensive survey that includes a description of the underlying principles and a discussion of the advantages and shortcomings of the various methods has been prepared by Willsky [9]. With such a wealth of background information available we shall forgo a detailed review of previous work in FDI. Instead, we proceed to the basic structure of a FDI system and the issues that require careful consideration during the design of such a system.

The FDI process can be thought of conceptually as consisting of two stages: residual-generation and decision-making. For a particular set of hypothesized failures, a general FDI system has the basic structure shown in Figure 1-1. Outputs from sensors are initially processed to enhance possible hypothesized failure effects so that they can be easily recognized. The processed measurements are called the residuals, and this enhanced effect of a failure is called the signature of the failure. Intuitively, the residuals represent the difference between the observed sensor outputs and the expected sensor outputs in the normal mode. In the absence of a failure, the residuals should be unbiased, showing agreement between observed and expected

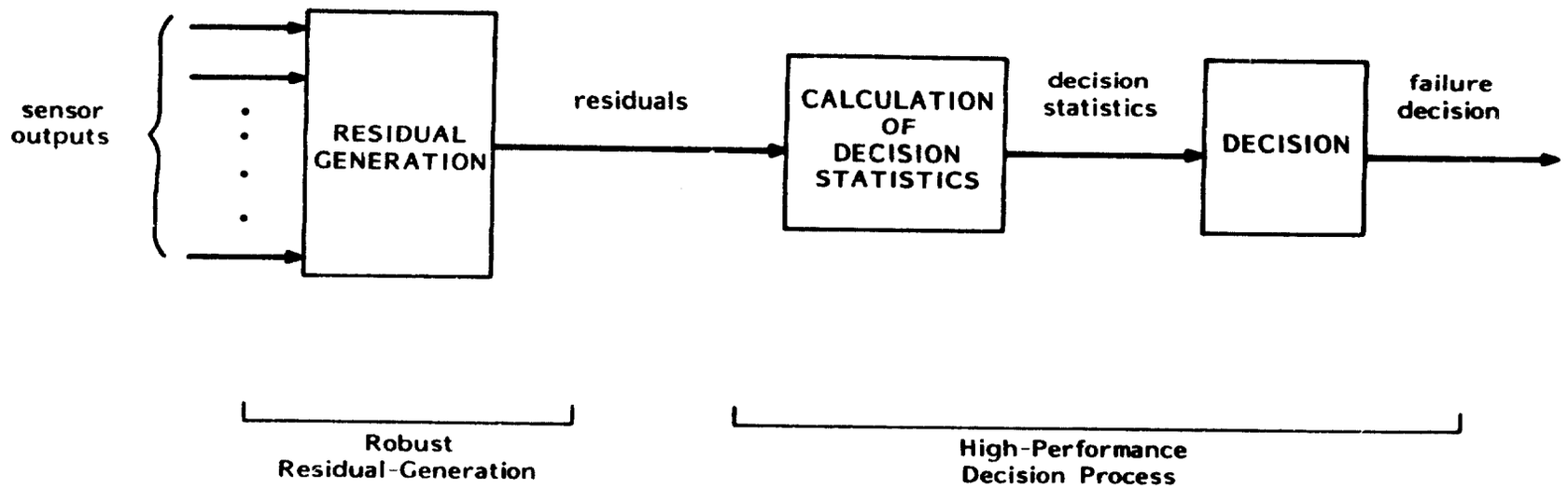


FIGURE 1-1: Two-stage Structure of the FDI Process.

normal behavior of sensor outputs, and a failure signature often takes the form of residual biases that are characteristic of the failure. The residual-generation process can be of varying degrees of complexity for different types of FDI systems. For example, in a voting system, the residuals are simply the differences of the outputs of the various pairs of like sensors, whereas in the GLR system, the residuals (which are also the filter residuals) are generated by the more complex KF.

In the decision process the residuals are examined for the presence of failure signatures. Decision functions or statistics are first calculated using the residuals. Then, a decision rule is applied to the decision statistics to determine if any failure has occurred in the system. A decision process may consist of a simple threshold test on the instantaneous values or moving averages of the residuals, or it may be based on more sophisticated from statistical decision theory e.g. the sequential probability ratio test (SPRT) [10].

The design of a FDI system requires the consideration of several issues. The immediate concern is the performance of the detection system, i.e. how responsive the system is to failures and how accurate the decisions are. Unfortunately, systems that respond quickly to abrupt changes are necessarily sensitive to noise effects. Thus, a tradeoff exists between detection speed and detection accuracy. In addition, the detection probabilities, i.e. the probabilities of correct detections and cross-detections (declaring one type of failure, when, in fact, another has occurred) cannot be arbitrarily

specified as parameters of the design. They represent additional tradeoffs inherent in the FDI design problem. These performance tradeoff issues can be most directly considered in the design of the decision process of the FDI system rather than the residual generation process. One of the goals of this research is to develop an approach for designing decision processes that systematically examines the tradeoffs among the various performance issues.

A desirable and important quality for a practical FDI system to possess is robustness, i.e. the relative insensitivity of the system's performance to parameter variations and modelling errors or uncertainties. An ideal approach to designing a robust system is to include all uncertainties in the problem specification, and a robust design will result from optimizing (in some sense) the performance of the system with the uncertainties. However, this generally leads to a complex mathematical problem that is too difficult to solve from a practical point of view. At the other extreme, a simpler alternative approach is to ignore all modelling uncertainties in the performance optimization process. The resulting design is then evaluated in the presence of modelling errors. If the degradation in performance is tolerable, the design is accepted, otherwise, it is modified and re-evaluated. Although this iterative method often yields acceptable designs, it has several serious drawbacks. Since the effects of the uncertainties are not directly determined, it is often unclear what parts of the design should be modified and what form the modifications should take. Furthermore, each iteration may be very expensive to carry out since extensive Monte Carlo simulations are often required for performance evaluation.

A better approach that considers the possible modelling errors directly is suggested in the work of Deckert, et.al. on aircraft sensor FDI problems [11]. The basic idea there is to identify the parts of the system that are known well and those that may contain substantial uncertainties. Then a FDI system (i.e. its residual-generation stage) is designed based primarily on the well-known parts (and only secondarily on the less well-known parts) of the system behavior. For example, the velocity and acceleration of an aircraft are related in two ways. Aerodynamic forces that give rise to the aircraft's acceleration are functions of the velocity (and other variables). However, this function relating velocity and acceleration is only known empirically and can be rather inaccurate. On the other hand, the kinematical relationship between velocity and acceleration is governed by a well-known physical relationship, $\dot{v}=a$. Therefore, the performance of a design based on the kinematical relationship is insensitive to system parameter variations, while a design based on the aerodynamics is sensitive to such variations.

Because, modelling errors affect the residual-generation process directly, the above approach suggests that robustness can effectively be achieved by designing a robust residual-generation process. We will adopt this approach to the robustness issue, as it is much simpler than the ideal approach (of an integrated treatment of robustness in the residual- and decision-making systems) and more direct than the trial-and-error method. Consequently, it will yield more insight into the general problem of robust FDI system design. In addition, this approach will provide the designer with a qualitative measure of the attainable level of robustness in the early stages of this design, and this will allow him to assess what be can expect in terms of overall performance.

Computational complexity is another important design consideration. Clearly, a practical system should only require a reasonable amount of storage and computation. An FDI system that take into account detailed dynamical behavior of the system is more complex but is likely to be more effective for a greater variety of failures than a system that does not use the same information. Furthermore, it may permit a reduction in hardware redundancy. In this study, the tradeoff effects among complexity, performance, and possible hardware redundancy are considered in the design of both the residual-generation and decision processes.

The goal of this research is to develop a methodology for designing FDI systems that takes into consideration the issues of performance, robustness, and computational complexity. Viewing the FDI process as consisting of two stages allows us to break up the FDI system design problem into two parts. We will examine the design of robust residual-generation processes and the design of high-performance decision-making proceeds separately.

1.2 Overview of Thesis

This thesis report basically consists of two parts, each dealing with one of the two stages of the FDI process. In Chapter 2 and 3 we will consider the design of residual-generation processes. The decision rule (decision process) design problem is the subject of Chapters 4,5, and 6.

All residual-generation processes exploit some form of analytical redundancy - the relationship among sensor outputs and actuator inputs specified by the dynamics of the system under the no-fail situation, e.g.

the kinematic relation $\dot{v}=a$. (when such a relation is violated, a failure among the components, i.e. sensors and actuators, involved in the relation must have occurred). In order to facilitate the design of robust residual-generation processes, a thorough understanding of the concept of analytical redundancy and how it can be exploited in deriving residuals is needed. In Chapter 2 we will present a characterization of analytical redundancy (for a linear time-invariant system in the absence of noise and modelling uncertainties) in terms of the concept of a parity space. We will describe several forms of residual-generation that are based on analytical redundancy, and we will discuss how such residuals can be used for FDI.

In Chapter 3 we will consider the effect of modelling errors and noise on redundancy relations, and we will define a simple measure of such effects. Clearly, a residual-generation process is robust (or as robust as it can be) if it is based on the redundancy relation that is least vulnerable to noise and modelling errors. The choice of such a redundancy relation is formulated as a minimax optimization problem (aimed at minimizing the worst case effect of noise and modelling error). Together with the viewing of analytical redundancy in terms of a parity space, the minimax design represent a new approach to the problem of designing robust residual-generation processes.

The design of a decision process involves resolving the tradeoff among detection performance issues such as expected detection delay, false alarm rates, and the various detection probabilities. We have chosen to examine this problem using the Bayesian approach with which the design problem can be

easily conceptualized. In Chapter 4, we will describe the Bayes formulation of the FDI decision problem and the optimal Bayes decision rule. Although the optimal rule is generally not computable, the structure of the Bayesian approach can be used to derive practical suboptimal rules. We will consider the design of suboptimal rules based on the Bayes formulation in Chapter 5. Numerical algorithms for designing such rules and evaluating the associated performance indices (detection probabilities, etc.) will also be presented. In Chapter 6, we will report on our experience with this approach to designing decision rules through a numerical example and simulation.

A brief summary of this thesis and a discussion of some future research directions are included in Chapter 7.

CHAPTER 2

ANALYTICAL REDUNDANCY AND RESIDUAL GENERATION

2.1 Introduction

The first stage of the FDI process is the generation of residuals, and one of the goals of this research is to investigate the problem of designing simple and robust residual-generation processes. To date, this problem has not been dealt with directly for the general case, although it was successfully resolved for a particular application [11]. The present chapter and the next one are devoted to developing one approach to this general design problem.

The first step towards our goal is to gain a better understanding of the concept of analytical redundancy - the basis for residual-generation. There are basically two forms of analytical redundancy: 1) direct redundancy- the instantaneous relationship among outputs of sensors, and 2) temporal redundancy - the relationship among the histories of sensor outputs and actuator inputs. Before we proceed to present a mathematical characterization of redundancy we will describe some examples of the two forms of redundancy.

Direct redundancy exists among sensors whose outputs are algebraically related, i.e. the sensor outputs are related in such a way that the variable one sensor measures can be determined by the instantaneous outputs of the other sensors. A simple example is the case of identical sensors, where we have, in the absence of sensor noise,

$$y_1 = y_2 \tag{2-1}$$

The general form of direct redundancy exists among a set of sensors that are modelled by linearly dependent c_j's in equation (1-2), e.g. a set of four accelerometers measuring acceleration in 3-space [12]. In such a case some fixed linear combination of the sensor outputs should always be zero (or close to zero when noise effects are included) in the normal mode. Alternatively, the ideal output of one of those sensors can be generated by a linear combination of the outputs of the remaining sensors, i.e.

$$y_1 = \sum_{i=2}^{m} \alpha_i y_i \qquad (2\text{-}2)$$

where the α_i are constants. It is clear that the identical sensor case (2-1) is a specialization of (2-2). In the absence of a failure, the ideal output calculated in this way should agree with the observed output of the sensor. That is, the <u>residual</u> $y_1(k) - \sum_{i=2}^{m} \alpha_i y_i(k)$ should be zero. A deviation from this behavior provides the clue to a failure among the set of sensors. We note that, through direct redundancy, certain dissimilar sensors may, in effect, be compared.

Direct redundancy has been exploited to generate residuals for the voting scheme for sensor FDI, where the "majority rule" principle is applied to detect and identify the failed sensors. (We will discuss voting in Section 2.3). Examples of successful application of the voting method include [2] [3][12]. The residuals of the voting system simply consist of weighted sums of sets of linearly dependent (instantaneous) sensor outputs. Thus, direct redundancy based residual-generation is simple. However it has

two major disadvantages. A high degree of hardware redundancy is required to use the "majority rule" principle. In addition, direct redundancy is not applicable for detecting actuator failures.

In contrast to direct redundancy, temporal redundancy is useful for both sensor and actuator FDI. Consider, for example, the temporal relationship between velocity (v) and acceleration (a):

$$v(k+1) = v(k) + Ta(k), \qquad k=1,2,... \tag{2-3}$$

where T is the period of discretization. Just as direct redundancy (2-2) provides the basis for comparing outputs of linearly dependent sensors, (2-3) prescribes a way of comparing velocity measurements with accelerometer outputs, i.e. the residual is $r(k+1) = v(k+1) - v(k) - Ta(k)$. As a result, outputs from velocity sensors and accelerometers can be compared in a mixed velocity-acceleration sensor voting system for detecting and identifying both types of sensor failures.

Temporal redundancy facilitates the comparison of sensor outputs among which direct redundancy does not exist. Consequently, a reduction in hardware redundancy for sensor FDI can be realized. Viewed in a different light, the use of analytical redundancy implies that additional sensor failures can in principal be detected with the same level of hardware redundancy.

To see how temporal redundancy can be exploited for detecting <u>actuator</u> failures, let us consider the first-order model of a vehicle in motion:

$$v(k+1) = \alpha v(k) + Tu(k), \quad k=1,2,... \tag{2-4}$$

where v denotes the vehicle's velocity, and α is a scalar constant between zero and one reflecting the effects of friction and drag. T is the discretization step, and u is the commanded engine force (actuator input) divided by the vehicle's mass. Now, the velocity measurements can be compared to the actuator inputs by means of (2-4), i.e. $r(k+1) = v(k+1) - \alpha v(k) - Tu(k)$. An actuator failure may be inferred, if the sensor is functioning normally but (2-4) is not satisfied.

While the additional information supplied by dissimilar sensor outputs and actuator inputs at different times through temporal redundancy facilitates the detection of a great variety of failures and reduces hardware redundancy, exploitation of this additional information often results in increased computational complexity, since the dynamics of the system will have to be accounted for. Depending on the accuracy of the system model, the biggest drawback, however, could be the increased sensitivity to system parameter variations due to the dependence on the system dynamics - the robustness issue.

From the above discussion, one approach to the design of robust residual-generation processes in any given application is evident: the various redundancies that are relevant to the failures under consideration are to be identified, and residual-generation should be based on the redundancies that are least sensitive to modelling uncertainties. This is the approach we will examine.

In order to apply this design philosophy, we need: 1) a precise characterization of analytical redundancy, and 2) a quantitative description

of the effects of noise and modelling uncertainties on the generation of residuals. We will examine the first problem in this chapter and the second problem in Chapter 3.

In the next section, we will present a general formulation of the concept of analytical redundancy in linear time-invariant systems. This formulation is a generalization of the parity equations studied by various researchers (e.g. [2],[3]) and the parity space discussed by Potter and Suman [13], and it provides a unified setting for discussing all approaches to FDI. In Section 2.3 we will discuss a generalized voting scheme for FDI, where residual-generation is based on the explicit forms of analytical redundancy described in section 2.2. In section 2.4 we will examine the effects of failures on the residuals generated from these explicit forms of analytical redundancy in order to understand how such information is used to detect failures in FDI schemes other than the voting method. In FDI systems such as GLR [4] and the detection filters of Beard [7] and Jones [8], residual-generation is accomplished by means of filters, which do not utilize analytical redundancy in as explicit a form as in a voting system. Based on the insights obtained in section 2.4 we will explore the role of analytical redundancy in the residual-generation process of these systems in section 2.5.

2.2 Analytical Redundancy - Parity Relation

In this thesis we have focussed our effort on developing an approach to designing robust residual-generation processes for linear time-invariant (LTI) systems. In order to focus on the concept of redundancy we will base our discussions in this chapter primarily on an LTI system that is in a

noise-free environment and for which we have an exact model. We will analyze the effect of noise in subsequent chapters as we consider the use of redundancy for the design of high-performance FDI schemes.

The system of interest to us here is characterized by the deterministic model

$$x(k+1) = Ax(k) + \sum_{j=1}^{q} b_j u_j(k) \tag{2-5}$$

$$y_j(k) = c_j x(k), \qquad j=1,\ldots,m \tag{2-6}$$

where x is the n-dimensional state vector, A is a constant nxn matrix, b_j is a constant (column) n-vector, and c_j is a constant (row) n-vector. The scalar u_j is the known input to the j-th actuator, and the scalar y_j is the output of the j-th sensor.

In order to facilitate the following discussion we introduce the following notation:

$$C_j(n_j) = \begin{bmatrix} c_j \\ c_jA \\ \vdots \\ c_jA^{n_j} \end{bmatrix} \qquad \begin{matrix} n_j = 0,1,\ldots \\ j = 1,\ldots,m \end{matrix} \tag{2-7}$$

The well-known Caley-Hamilton theorem [15] implies that there is an $\bar{n}_j$, $1 \le \bar{n}_j \le n$, such that

$$\text{rank } C_j(n_j) = \begin{cases} n_j+1 & n_j < \bar{n}_j \\ \bar{n}_j & n_j \ge \bar{n}_j \end{cases} \tag{2-8}$$

The matrix $C_j(\bar{n}_j-1)$ characterizes that part of the system that is observable from the j-th sensor. Specifically, the null space of $C_j(\bar{n}_j-1)$, $N(C_j(\bar{n}_j-1))$, is known as the unobservable subspace of the j-th sensor, because any component of the state lying in $N(C_j(\bar{n}-1))$ will not affect the output of the j-th sensor [15]. The rows of $C_j(\bar{n}_j-1)$ span a subspace of R^n that is the orthogonal complement of the unobservable subspace. Such a subspace is defined here to be the observable subspace of the j-th sensor, and it has dimension $\bar{n}_j$. The system (2-5)-(2-6) is observable (through the m sensors) if the sum of the m observable subspaces is the whole space R^n. We will assume that the system is observable.

In Subsection 2.2.1 we will characterize analytical redundancy in terms of the concept of a parity space and parity relations, and in Subsection 2.2.2 we will discuss residual-generation schemes based on analytical redundancy, i.e. on parity relations.

2.2.1 The Generalized Parity Space

Let ω be a row vector of dimension $\bar{n} = \sum_{j=1}^{m} (\bar{n}_j+1)$ such that $\omega = [\omega^1 \ldots \omega^m]$, where ω^j, $j=1,\ldots,m$, is a $(\bar{n}_j+1)$-dimensional row vector. Consider a non-zero ω satisfying

$$[\omega^1,\ldots\omega^m] \begin{bmatrix} C_1(\bar{n}_1) \\ \vdots \\ C_m(\bar{n}_m) \end{bmatrix} x = 0, \qquad \forall x \in R^n \tag{2-9}$$

(Note that in the above equation $C_j(\bar{n}_j)$ has $\bar{n}_j+1$ rows while it has only rank $\bar{n}_j$. The reason for this will become clear when we discuss the temporal

redundancy associated with a single sensor.) Since the system is observable, there are only $\bar{n}$-n linearly independent ω's that satisfy (2-9). We let Ω be an $(\bar{n}-n)\times\bar{n}$ matrix with a set of such independent ω's as its rows. (The matrix Ω is not unique.) Assuming all the u_j's are zero for the moment, we have the $\bar{n}$-n linearly independent parity equations or parity relations that are independent of the state x:

$$\Omega \begin{bmatrix} y_1(k,\bar{n}_1) \\ \vdots \\ y_m(k,\bar{n}_m) \end{bmatrix} = p \qquad (2\text{-}10)$$

where

$$y_j(k,\bar{n}_j) = \begin{bmatrix} y_1(k) \\ \vdots \\ y_j(k+\bar{n}_j) \end{bmatrix} \quad j=1,\ldots,m \qquad (2\text{-}11)$$

The $(\bar{n}-n)$-vector p is called the parity vector. Under the ideal conditions set forth in the beginning of this section, p is zero. More generally, in the presence of noise and failures, p is a non-zero vector representing the inconsistencies among the sensor outputs. Different failures will produce different p's. Thus, the parity vector may be used as the signature-carrying residual for FDI. We will further discuss residual-generation based on parity equations in the succeding sections.

The space of all $(\bar{n}-n)$-dimensional parity vectors defined by (2-10) is called a parity space. We note that the parity space discussed above is an extension of the parity space examined by Potter and Suman [13] to

include sensor outputs at different times, thereby, taking into account the dynamics of the system. In [13], Potter and Suman exclusively studied equation (2-10) with $\bar{n}_1=\bar{n}_2= \ldots =\bar{n}_m=0$. We will exploit our generalized notion of a parity space to characterize analytical redundancy.

When the actuator inputs are not zero, (2-10) must be modified to be

$$\Omega\left\{\begin{bmatrix} y_1(k,\bar{n}_1) \\ \vdots \\ y_m(k,\bar{n}_m) \end{bmatrix} - \begin{bmatrix} \mathcal{B}_1(\bar{n}_1) \\ \vdots \\ \mathcal{B}_m(\bar{n}_m) \end{bmatrix} U(k,\bar{n}_0)\right\} = p \qquad (2\text{-}12)$$

where

$$\mathcal{B}_j(\bar{n}_j) = \begin{bmatrix} 0 & & & & & & \\ & \ddots & & & & 0 & \\ & & \ddots & & & & \\ c_jB & & & \ddots & 0 & & \\ & & & & \ddots & & \\ \vdots & & & & & \ddots & \\ & & & & & & \ddots \\ c_jA^{\bar{n}_j-1}B & \ldots & c_jB & 0 & \ldots & 0 & \end{bmatrix} \qquad (2\text{-}13)$$

$$B = [b_1 \ldots b_q] \qquad (2\text{-}14)$$

$$\bar{n}_0 = \max[\bar{n}_1,\ldots,\bar{n}_m] \qquad (2\text{-}15)$$

$$u(k) = [u_1(k) \ldots u_q(k)]' \qquad (2\text{-}16)$$

$$U(k,\bar{n}_0) = [u'(k) \ldots u'(k+\bar{n}_0)]' \qquad (2\text{-}17)$$

$\mathcal{B}_j(\bar{n}_j)$ is an $(\bar{n}_j+1)\times \bar{n}_0 q$ matrix (q is the number of actuators). Equation (2-12) defines the <u>generalized parity vector</u> p, and the $(\bar{n}-n)$-dimensional

space of all such vectors is called the generalized parity space. Any linear combination of the rows of the left hand side of (2-12) is called a parity function. Note that (2-8) implies that we generally do not need to consider a higher dimensional parity space that is defined by (2-12) with $\bar{n}_j$ replace by $n_j > \bar{n}_j$, j=1,...,m, although it is possible to do so

It is now clear that (2-12) with p=0 (which is the case under ideal conditions) characterizes all the analytical redundancies of the LTI system (2-5) and (2-6), because it specifies all the essential relationships among the actuator inputs and sensor outputs. Each parity equation can be regarded as a redundancy relation, and it can be obtained by taking a linear combination of the rows of (2-12).

An important notion in describing analytical redundancy is the order of a redundancy relation. Let ω be the vector of a particular parity relation, i.e.

$$\sum_{j=1}^{m} \omega^j [Y(k,\bar{n}_j) - B_j(\bar{n}_j)U(k,\bar{n}_0)] = 0 \qquad (2\text{-}18)$$

where $[\omega^1 \ldots \omega^m] = \omega$. We can define the order ρ of such a relation as follows. Since some elements of ω may be zero, there is a largest index $\tilde{n}$ such that the $\tilde{n}$-th element of ω^j for some j is non-zero but the $(\tilde{n}+1)$th through the $\bar{n}_j$-th elements of each ω^j are zero (or $\bar{n}_j \leq \tilde{n}$.) Then, ρ is defined to be $\rho=\tilde{n}-1$. The order ρ describes the "memory span" of the redundancy relation. For example, when $\rho=0$, instantaneous outputs of sensors are examined. When $\rho>0$, at least some sensor outputs at times up to ρ steps in the past need to be considered in the parity equation, e.g. the kinematical equation (2-3)

is a first order parity relation. Hence, direct redundancy is characterized by $\rho=0$, while a temporal redundancy relation has a $\rho>0$.

Based on the properties of observability subspaces we have developed a general characterization of analytical redundancy in terms of a parity space. To illustrate the generality of this characterization we will examine a few examples of redundancy relations.

Direct Redundancy

Direct redundancy is described by a zeroth order parity relation

$$[\omega_0^1\ 0\ldots0|\ldots|\omega_0^m\ 0\ldots0]\left\{\begin{bmatrix} y_1(k,\bar{n}_1) \\ \vdots \\ y_m(k,\bar{n}_m) \end{bmatrix} - \begin{bmatrix} \mathcal{B}_1(\bar{n}_1) \\ \vdots \\ \mathcal{B}_m(\bar{n}_m) \end{bmatrix} U(k,\bar{n}_0)\right\} = 0 \qquad (2\text{-}19)$$

where ω_i^j is a scalar denoting the (i+1)-st elements of ω^j. At least two of the ω_i^j must be non-zero for (2-19) to be a meaningful parity relationship. Because of the structure of $\mathcal{B}_j(\bar{n}_j)$ (see (2-13)), (2-19) can be written as

$$[\omega_0^1\ldots\omega_0^m]\begin{bmatrix} y_1(k) \\ \vdots \\ y_2(k) \end{bmatrix} = 0 \qquad (2\text{-}20)$$

In this case the parity function (left hand side of (2-20)) can be directly used as the residual.

A single sensor

Due to (2-8) (Caley-Hamilton) it is always possible to find a non-zero ω^j such that under the ideal condition, (2-12) becomes

$$\omega^j [Y_j(k,\bar{n}_j) - B_j(\bar{n}_j)U(k,\bar{n}_0)] = 0 \qquad (2\text{-}21)$$

Expression (2-21) represents a form of temporal redundancy - i.e. it is the relationship among the histories of the j-th sensor output and the actuator input, and it is of order $\bar{n}_j$. Note that (2-21) is the lowest order parity relation involving only one (the j-th) sensor, and this is why we have chosen to consider $C_j(\bar{n}_j)$, as opposed to $C_j(\bar{n}_j-1)$, in defining the generalized parity space. Parity relation (2-21) prescribes a consistency test that requires comparing to zero a linear combination of a window of sensor j outputs and the actuator inputs. Such a combination (the left hand side of (2-21)) can be used as the residual r(k). Since this test involves only one sensor, it may be used as a self-test for sensor j, if $B_j(\bar{n}_j)=0$ or if the actuators can be verified (by other means) to be functioning properly. Similarly, it can be used to detect actuator failures when sensor j can be verified to be normal. The Caley-Hamilton theorem implies that a self-test redundancy such as (2-21) always exists for sensor j.

Equation (2-21) can be alternatively written as

$$y_j(k) = -(\omega^j_{\bar{n}_j})^{-1}\left[\sum_{t=1}^{\bar{n}_j}\omega^j_{\bar{n}_j-t}\, y_j(k-t) - \sum_{t=1}^{\bar{n}_j}\sigma^j_{\bar{n}_j-t}\, u(k-t)\right] \qquad (2\text{-}22)$$

where

$$[\sigma_0^j \ \ldots \ \sigma_{\bar{n}_j-1}^j \ 0\ldots0] = \omega^j \beta_j(\bar{n}_j) \tag{2-23}$$

and σ_t^j, $t=0,\ldots,\bar{n}_j-1$, is a q-dimensional row vector; ω_t^j, $t=0,\ldots,\bar{n}_j$ is the (t+1)-th component of ω^j, and u(k) is the q-dimensional actuator input vector at time k. Equation (2-22) represents an auto-regressive moving-average (ARMA) model for the j-th sensor output. It is only a moving average (MA) if $c_j A^{\bar{n}_j}=0$. Under the ideal condition, the value of y_j at time k can be predicted from the past values of y_j and actuator inputs using (2-22). The residual defined by taking the difference between the left and right hand sides of (2-22) is indeed the difference between such a prediction of $y_j(k)$ and the observed $y_j(k)$. Hence, a non-zero residual will provide the clue for a (sensor j or an actuator) failure.

Temporal redundancy between two sensors

A temporal redundancy exists between sensor i and sensor j, if there are

$$\omega^i = [\omega_0^i \ldots \omega_{\bar{n}_i-1}^i \ 0] \tag{2-24a}$$

$$\omega^j = [\omega_0^j \ldots \omega_{\bar{n}_j-1}^j \ 0] \tag{2-24b}$$

satisfying the redundancy relation

$$[\omega^i \omega^j] \left\{ \begin{bmatrix} y_i(k,\bar{n}_i) \\ y_j(k,\bar{n}_j) \end{bmatrix} - \begin{bmatrix} \beta_i(\bar{n}_i) \\ \beta_j(\bar{n}_j) \end{bmatrix} \ u(k,\bar{n}_0) \right\} = 0 \tag{2-25}$$

Equation (2-25) is a special case of the general form of parity equation (2-18) with $\omega^s=0$, $s\neq i$, $s\neq j$. The relation (2-25) is of order $\rho< \max[\bar{n}_i,\bar{n}_j]$. Clearly, (2-25) holds if

$$[\omega_0^i \ldots \omega_{\bar{n}-1}^i]\, C_i(\bar{n}_i-1) = -[\omega_0^j \ldots \omega_{\bar{n}_j-1}^j]\, C_j(\bar{n}_j-1) \qquad (2\text{-}26)$$

Now, the rows of $C_i(\bar{n}_i-1)$ and $C_j(\bar{n}_j-1)$ span the observable subspaces of sensors i and j, respectively. Hence, (2-26) implies that a redundancy relation exists between two sensors if their observable subspaces overlap. Furthermore, when the overlap subspace is of dimension $\hat{n}$, there are $\hat{n}$ linearly independent $[\omega^i\omega^j]$ pairs that will satisfy (2-26). Therefore, there are as many independent redundancy relations of the form (2-25) as the dimension of the overlap subspace.

Because the order of the redundancy (2-25) is ρ, either ω_ρ^i or ω_ρ^j must be non-zero. Assuming $\omega_\rho^j\neq 0$, we can write (2-25) in the form of an ARMA model for y_j:

$$y_j(k) = -(\omega_\rho^j)^{-1}\left[\sum_{t=1}^{\rho}\omega_{\rho-t}^j y_j(k-t) + \sum_{t=0}^{\rho}\omega_{\rho-t}^i y_i(k-t) + \sum_{t=1}^{\rho}(\sigma_{\rho-t}^j+\sigma_{\rho-t}^i)u(k-t)\right] \qquad (2\text{-}27)$$

where we have used the notation (2-23). (Note that the summation of the i-th sensor outputs ranges from 0 to ρ but those of the j-th sensor outputs and actuator inputs are from 1 to ρ.) Note that viewing (2-27) as an ARMA model for y_j, we see that y_i plays the role of an input, just as do the actuator inputs u.

Similar to the single sensor case, (2-27) indicates that $y_j(k)$ can be predicted from a linear combination of the i-th sensor outputs $(y_i(k-\rho),\ldots,y_i(k))$, the j-th sensor outputs $(y_j(k-\rho),\ldots,y_j(k-1))$, and the actuator inputs $(u(k-\rho),\ldots,u(k-1))$. The kinematical relation between velocity and acceleration measurements (2-3) is a parity relation expressed in the form of (2-27). The parity function (the left hand side of (2-25)), which represents the difference between the observed and predicted sensor outputs, can be directly used as the residual.

In summary, we have conceptualized the notion of analytical redundancy in terms of a generalized parity space. We have also illustrated how various redundancy relations can be obtained from this parity space and how these relations may be used in forming residuals. In the next subsection we will further discuss residual-generation based on parity relations.

2.2.2 Residual Generation Based on Parity Relations

In the preceding discussion we saw that parity functions can be used as residuals. These residuals may in turn be used in a voting system for FDI. We will discuss the voting scheme in the next section. In the remainder of this section we will describe other methods for generating residuals based on parity relations (temporal redundancies). We will mainly use the kinematical relation (2-3), which is a first order parity relation involving two sensors, to illustrate these mechanisms of residual-generation, but the basic concept can be readily generalized to higher order cases involving more sensors and actuators.

For easy reference we re-write (2-3) here

$$v(k+1) = v(k) + Ta(k) \tag{2-28}$$

A direct method for generating a residual $r^1(k)$ is as follows

$$r^1(k) = v(k) - v(k-1) - Ta(k-1) \tag{2-29}$$

This is an example of the type of residual described in the preceding section that involves direct calculation of a parity function as in equation (2-18). In a noisy environment, $r^1(k)$ is a random sequence. In the absence of a failure it is zero mean. When a failure occurs it becomes biased (possibly for only a short period of time as we shall see below). It is by detecting the presence of the bias in $r^1(k)$ that a failure can be inferred. Now consider a velocity sensor failure that manifests itself as a constant bias in the v-measurement. Suppose this failure occurs at time τ. Then $r^1(k)$ will contain a bias at time τ, but it will become zero mean for $k>\tau$. That is, the failure signature vanishes after one time step. (This is because the sensor bias effect is cancelled out via the term: $v(k) - v(k-1)$ in (2-29)). Thus, if this failure is not detected at time τ, $r^1(k)$ defined by (2-29) will not provide any clue of this failure after time τ.

Fortunately, another way of using the parity relation (2-29) to generate useful residuals is available. The ARMA representation (2-29) of the kinematical relation implies that with an initial observation of the v-measurement, say $v(0)$, $v(k)$, $k=1,2,..$ can be predicted using the acceleration measurements only. Such a predicted $v(k)$ can be subtracted from the

observed v(k) to form the residuals $r^2(k)$:

$$\tilde{v}(k+1) = \tilde{v}(k) + Ta(k) \tag{2-30a}$$

$$r^2(k) = v(k) - \tilde{v}(k) \tag{2-30b}$$

where v and $\tilde{v}$ denote the observed and the (open-loop) predicted velocity measurements, respectively. Since no velocity measurement is used in the prediction other than during the initialization, the bias effect of the failure (its signature) will be present in $r^2(k)$ for $k \geq \tau$. In addition, a constant accelerometer bias will produce a ramp in r^2 but only a constant bias in r^1. The possible drawback of this scheme is that noise effects (due to the accelerometer) are accumulated. Therefore, it is useful for cases where the noise accumulated over the failure-monitoring period is small. Alternatively, when the noise level is low, this scheme can be applied with periodic re-initialization of the velocity prediction process. A variation of this residual-generation scheme for accelerometer failures was used with success in the aircraft sensor FDI problem [11].

A third residual-generation scheme may be devised using the parity relation (2-29) in a closed-loop fashion. Based on the ARMA representation (2-29) a filter for the velocity can be constructed:

$$\hat{v}(k+1) = \hat{v}(k) + Ta(k) + hr^3(k) \tag{2-32a}$$

$$r^3(k) = v(k) - \hat{v}(k) \tag{2-32b}$$

where $\hat{v}$ denotes the closed-loop prediction of the velocity measurement, and h is the filter gain ($0<h<1$). The filter residuals $r^3(k)$ also represent

the difference between the observed and predicted (or expected) velocity measurement, and hence, can be used as residuals for FDI. The advantage of using r^3 over r^2 is that the filter gain can be chosen so that the variance of $\hat{v}$ (hence that of r^3) will not grow indefinitely with time. As a result, the periodic re-initialization of the prediction process can be eliminated. The tradeoff, for example, is that the signature contained in r^3 for a velocity sensor bias failure, will vanish with increasing elapsed time k-T, and that an accelerometer bias will lead to a steady state bias, not a ramp, in r^3. This residual-generation scheme is used in FDI schemes such as the GLR [4].

In summary, we have described three ways a temporal redundancy or parity relation may be used to generate residuals for FDI. (Note that for direct redundancy only the first method i.e. r^1, is available since no dynamics are involved). Generally, $r^1(k)$ is the residual of the <u>instantaneous</u> comparison of the left hand side and right hand side of the parity equation (2-27), and it is dependent on $y_j(k-\rho),\ldots,y_j(k)$. The residual $r^2(k)$ is the difference between $y_j(k)$ and its predicted value that is computed from $y_j(0),\ldots,y_j(\rho-1)$, $u(t)$, and $y_i(t)$, $i\neq j$, $t=1,\ldots,k$ using (2-27). Thus r^2 effectively represents a <u>dynamic</u> comparison (since all past u and y_i are used via the dynamics (2-27) in forming r^2). In contrast to $r^1(k)$, $r^2(k)$ depends on $y_j(0),\ldots,y_j(\rho-1)$, and $y_j(k)$ but not on $y_j(k-\rho),\ldots,y_j(k-1)$. The third type of residual $r^3(k)$ is the innovations of the filter of $y_j(k)$ based on (2-27). Similar to r^2, r^3 is based on a dynamic comparison. Moreover, $r^3(k)$ depends on $y_j(k-\rho),\ldots,y_j(k-1)$ just as $r^1(k)$, albeit in a closed-loop

manner. These residuals can be directly used in a voting scheme, which will be discussed in the next section. Other methods for exploiting the failure information contained in the parity space will be explored in Sections 2.4 and 2.5.

2.3 The Generalized Voting Scheme

In this section we will describe how parity relations can be utilized in a generalized voting scheme (GVS) for FDI. Other usage of parity relations for FDI will be discussed in the next two sections. For simplicity we will assume in this section that only one failure can occur, but extension of the following idea to the case of simultaneous failures is straightforward.

The structure of a generalized voting system based on M parity relations is shown in Figure 2-1. This FDI system basically consists of M tests each of which serves to determine if one of the M parity relations is violated. Each test has its own residual-generation process that is based on a single parity relation. If the underlying parity relation is a direct redundancy, the residual is simply the parity function. If the parity relation represents a temporal redundancy, then residual-generation may take on one of the three forms discussed in Subsection 2.2.2, i.e. the residual may be simply the parity function, the difference between the true observation and the open-loop prediction of a sensor output, or the difference between the true observation and the closed-loop prediction of a sensor output. A decision rule is applied to the residuals associated with

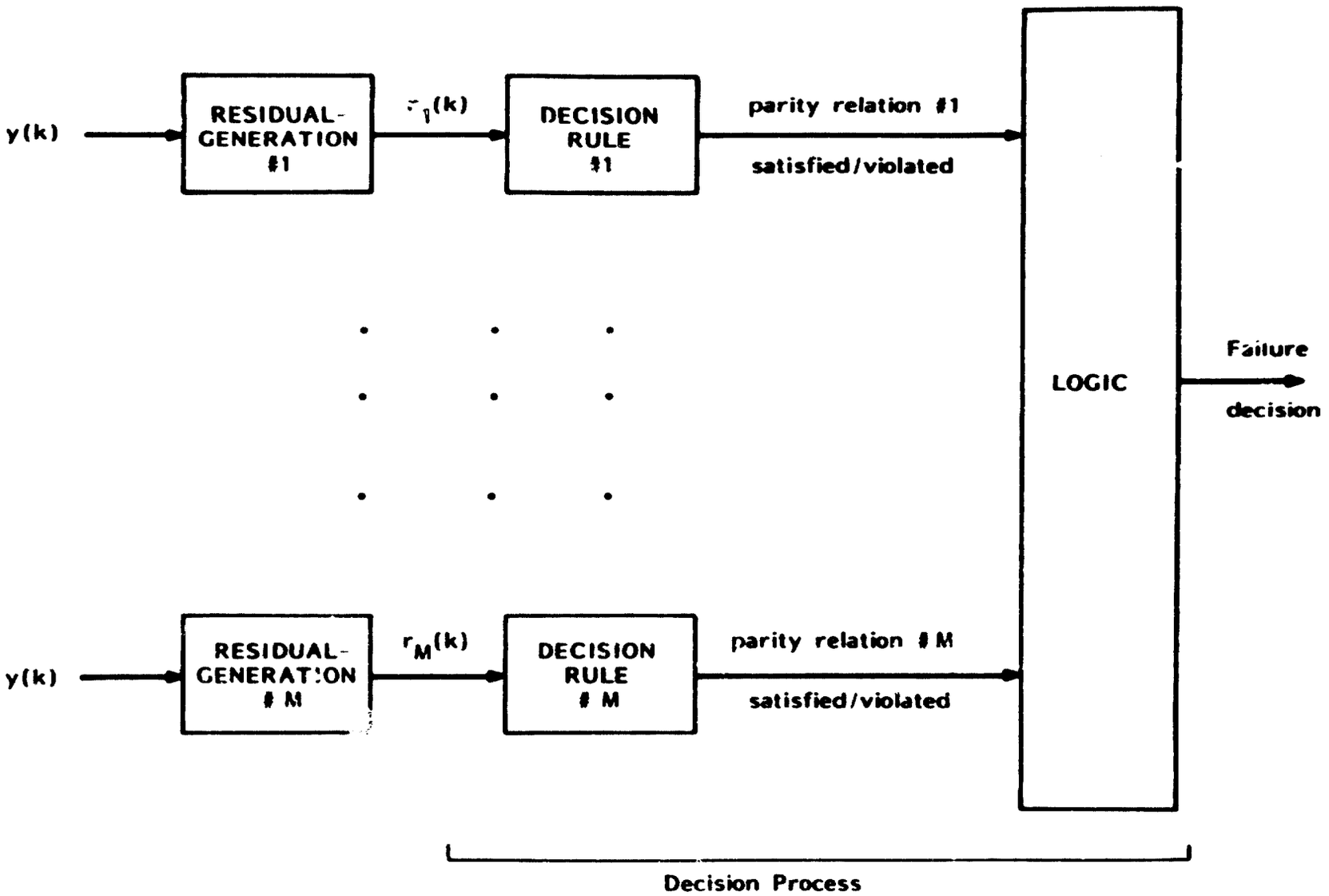


FIGURE 2-1: The Generalized Voting Scheme.

each parity relation to determine if the corresponding parity relation is violated. A different decision rule may be used for each of the M parity relations. Typically, the decision rule employed in a voting system takes the form of a threshold tests on a single point of residual or the moving average of a window of residuals. However, more sophisticated rules such as the sequential rules examined in Chapter 4,5, and 6 may be applied to make better use of the failure information contained in the residual for high-performance. At the last stage of the GVS, the voting logic (to be described in the following) is applied to the outcome of the M consistency tests to detect and identify the failed component. Next, we will describe the voting logic.

In order to apply GVS, we need a set of parity relations with the property that each component (i.e. a sensor or an actuator) of interest is included in at least one parity relation and each component is excluded from at least one of the parity relations. (If there are M components, the number of parity relations considered is M or more. However, later in this section we will see that with a slight modification of the logic described below, as few as M-1 parity relations can be used.) When a component fails, all the parity relations involving it will be violated, while those excluding it will still hold. This means that the components involved in parity relations that hold can immediately be declared as unfailed. Moreover, the one component that is common to all of the violated parity relations is then readily identified as failed. This is the basic idea of generalized voting and is also the logic used by GVS to detect and identify failed components. It differs from the common notion of voting in the sense that through analytical

redundancy (parity relations) dissimilar components (including sensor and actuators) may vote together. We note that the voting system for linearly dependent sensor studied by various researchers [2][3][12] and the aircraft sensor FDI system [11] are special cases of GVS.

In the remainder of this section we will discuss some important considerations involved in designing a generalized voting system. We will do so by means of a second order (n=2) example:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix} \tag{2-33a}$$

$$b = [0 \quad 1]' \tag{2-33b}$$

$$c_1 = [1 \quad 0] \tag{2-33c}$$

$$c_2 = [0 \quad 1] \tag{2-33d}$$

In this case, $\bar{n}_1=2$, $\bar{n}_2=1$, $\bar{n}-n=3$, and there are (only) three independent parity equations:

$$y_1(k)-(a_{11}+a_{22})y_1(k-1) + a_{11}a_{22}y_1(k-2) - a_{12}u(k-2)=0 \tag{2-34}$$

$$y_1(k) - a_{11}y_1(k-1) - a_{12}y_2(k-1)=0 \tag{2-35}$$

$$y_2(k) - a_{22}y_2(k-1) - u(k)=0 \tag{2-36}$$

These parity relations can be applied in a GVS for detecting failures in the sensors and the actuator, because each of the three terms y_1, y_2, and u

is included in two parity relations and excluded from one. Since all the parity relations represent temporal redundancy, any of the three forms of residual-generation (see Subsection 2.2.2) may be used. Three important issues concerning the design of a GVS are examined in the following.

1) The output y_2 appears in (2-35) and (2-36) with different time lags (and so does u in (2-34) and (2-36)). Therefore, a sensor 2 failure will violate the parity relation (2-35) one time step later than (2-36) (regardless of the form of residual-generation used). A decision process responding quickly to this effect will declare an actuator failure. But this is erroneous. If one more time step is considered, both parity relations would be violated, and the correct failed component (sensor 2) can be identified. Although this type of transient behavior will disappear (for open-loop residuals, it will disappear in less than n steps, where n is the dimension of the system), it suggests that temporal behavior of the residuals should be carefully considered in designing decision processes that can respond quickly and accurately.

2) Under the assumption that only one failure can occur, only two of the three parity relations (2-34)-(2-36) are needed for FDI. To see this, consider only (2-34) and (2-35). An actuator failure affects only (2-34) and a sensor 2 failure affects only (2-35), while a sensor 1 failure affects both (2-34) and (2-35). Thus, the voting logic can be modified to recognize these failure phenomena, and FDI can be accomplished based on two parity relations. In fact, it is easy to see that any combination of two of the

above three parity relations may be employed to generate residuals for use with the modified voting logic for FDI, and we have a choice among three combinations.* In the deterministic case with an exact model, all such combinations will serve equally well, and we may use any one of these combinations of parity relations. However, in the presence of noise and modelling uncertainties, all the parity relations are obscured, albeit to different extents. Thus, residual-generation should be based on parity relations that are least vulnerable to such adverse effects. This design approach is the focus of this part of our research and it will be fully considered in Chapter 3.

3) While some systems have more parity relations than needed for voting, others may have less than the necessary number. For instance, suppose in the above example we replace the single actuator with two actuators characterized by $b_1 = [1 \;\; 1]'$ and $b_2 = [-1 \;\; 1]$, respectively. This new configuration will not change $\bar{n}$ since A and c_j remain unchanged, and there are three parity relations:

$$y_1(k)-(a_{11}+a_{22})y_1(k-1)+a_{11}a_{22}y_1(K-2)+(a_{22}-a_{12})u_1(k-1)-u_1(k-2)-(a_{22}+a_{12})u_2(k-1)+u_2(k-2)=0 \qquad (2\text{-}37)$$

$$y_1(k) - a_{11}y_1(k-1) - a_{12}y_2(k-1) - u_1(k-1) + u_2(k-1)=0 \qquad (2\text{-}38)$$

$$y_2(k) - a_{22}y_2(k-1) - u_1(k-1) - u_2(k-1)=0 \qquad (2\text{-}39)$$

These three parity relations are inadequate

---

* A system may inherently have more candidate parity relations than required for GVS (with or without the modified logic). For example, suppose $C_2=[1 \;\; 1]$ in (2-33d). Then $\bar{n}_2=2$, $\bar{n}-n=4$, i.e. there are four independent parity equations, while only three (two for the modified logic) are needed.

above three parity relations may be employed to generate residuals for use with the modified voting logic for FDI, and we have a choice among three combinations.* In the deterministic case serve equally well, with an exact model all such combinations will and we may use any one of these combinations of parity relations. However, in the presence of noise and modelling uncertainties, all the parity relations are obscured, albeit to different extents. Thus, residual-generation should be based on parity relations that are least vulnerable to such adverse effects. This design approach is the focus of this part of our research and it will be fully considered in Chapter 3.

3) While some systems have more parity relations than needed for voting, others may have less than the necessary number. For instance, suppose in the above example we replace the single actuator with two actuators characterized by $b_1 = [1 \quad 1]'$ and $b_2 = [-1 \quad 1]$, respectively. This new configuration will not change $\bar{n}$ since A and C remains unchanged, and there are three parity relations:

$$y_1(k)-(a_{11}+a_{22})y_1(k-1)+a_{11}a_{22}y_1(k-2)+(a_{22}-a_{12})u_1(k-1)-(a_{22}+a_{12})u_2(k-1)+u_2(k-2) \tag{2-37}$$

$$y_1(k) - a_{11}y_1(k-1) - a_{12}y_2(k-1) - u_1(k-1) + u_2(k-1)=0 \tag{2-38}$$

$$y_2(k) - a_{22}y_2(k-1) - u_1(k-1) - u_2(k-1)=0 \tag{2-39}$$

Note that (2-39) is the same as (2-36). These relations are inadequate

---

* A system may inherently have more candidate parity relations than required for GVS (with or without the modified logic). For example, suppose $C_2=[1\ 1]$ in (2-33d). Then $\bar{n}_2=2$, $\bar{n}-n=4$, i.e. there are four independent parity equations, while only three (two for the modified logic) are needed.

for GVS, which requires four relations for four components. (These relations are also in-sufficient for use with the modified logic described earlier, because a sensor 1 and an actuator 2 failure will both violate all three parity relations.) Therefore, other FDI schemes that exploit the failure information carried by the residuals in ways different for GVS will have to be used. We will examine these methods in the next two sections.

2.4 Failure Characteristics in Parity Space

The generalized voting scheme discussed in the previous section represents one method for using one form of the failure information contained in the residuals for FDI. In this section we will examine other methods of exploiting this information to detect and identify failures, and we will contrast them with GVS. We will primarily consider (open-loop) residuals that are generated using parity functions, i.e. the residual vector is simply a parity vector. In Section 2.5 we will discuss the case with (closed-loop) residuals generated by filters.

First we will consider sensor FDI using direct redundancy. Based on direct redundancy, the residual vector r(k) has the form (y is m-dimensional)

$$r(k) = \bar{\Omega} y(k) \tag{2-40}$$

Each row of the right hand side of (2-40) is a parity function. For GVS, $\bar{\Omega}$ is chosen such that for each sensor (j) there is at least one component of r that is dependent on y_j and one component that is independent of y_j. When a sensor j failure occurs, all residual components depending on y_j will become non-zero, while all other components remain zero (assuming no noise).

The mechanism of the GVS involves examining each individual component separately for a bias. Based on the location of the biases in r, a failure identification is made.

Another way of using the failure information contained in the residuals is as follows. A faulty sensor, say the j-th one, contains an error signal $\nu(k)$ in its outputs

$$y_j(k) = c_j x(k) + \nu(k) \tag{2-41}$$

When this sensor fails, we have (assuming no noise)

$$r(k) = \bar{\Omega}_j \nu(k) \tag{2-42}$$

where $\bar{\Omega}_j$ is the j-th column of $\bar{\Omega}$. That is, no matter what $\nu(k)$ is, the effect of a sensor j failure on the residual always lie in the direction $\bar{\Omega}_j$. Thus, $\bar{\Omega}_j$ is the failure direction in parity space (FDPS) corresponding to sensor j. (In [13], $\bar{\Omega}_j$ is referred to as the j-th measurement axis in parity space.) If an $\bar{\Omega}$ can be chosen such that all its columns represent distinct directions in the parity space*, then a sensor of failure, j=1,...,m, can be inferred from the presence of a bias component in the residual along $\bar{\Omega}_j$. Clearly an Ω suitable for voting satisfies this condition. Generally, there may exist an $\bar{\Omega}$ with as few as two rows and with columns pointing in m distinct directions in the parity space. This approach to sensor FDI was studied by Potter and Suman [13] and Daley et al [16].

---

* The columns of Ω are, in fact, linearly dependent (but possible distinct) because there are at most $m-n_0$ linearly independent parity functions (rows of Ω) while there are m columns in Ω. Here, m is the number of sensors, and n_0 is the rank of C (C is a matrix with c_j as its rows, and n_0 is the number of linearly independent sensors).

A FDI system based on recognizing FDPS examines all the residual components together, because the failure signature being looked for is a particular direction in parity space not restricted to one of the coordinate axes defined by individual parity relations. This FDI scheme (called the FDPS method for brevity) is different from the GVS in that the latter examines each residual component separately. These two schemes actually exploit different aspects of the failure information carried by the residuals. This difference can be illustrated by a simple example described below. Suppose that the residual r(k) is a 3-vector given by (2-40). To provide a basis for comparing GVS with the FDPS method we assume that $\bar{\Omega}$ is chosen such that r is suitable for use in both FDI methods. Let us further suppose that the second component r_2 of the residual is independent of y_2. In order to detect a sensor 2 failure, GVS will look for a bias in r_1 and a bias in r_3 (see Figure 2-2a). To detect the same failure, the FDPS methods will search for a signature in the direction $\bar{\Omega}_2$ in the r_1-r_3 plane (see Figure 2-2b). In this case the FDPS is defined by a precise combination of the biases in r_1 and r_3, and it represents a more detailed characterization of the failure signature (information) than the biases considered by GVS. It is by the exploitation of this detailed information that the FDPS method can, at least in theory, detect and identify m sensor failures using a 2-dimensional r, i.e. $\bar{\Omega}$ has two rows but m distinct columns.

These two forms of failure information are also used by GVS and the FDPS method when temporal redundancy is employed. In such cases GVS still examines each residual component separately for the presence of a bias.

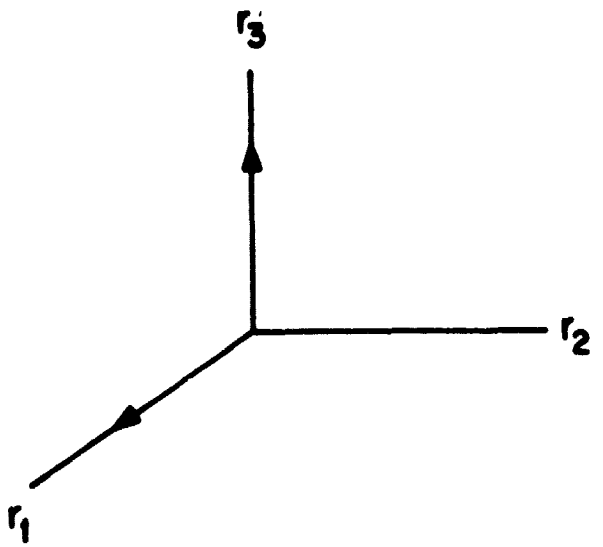


FIGURE 2-2a: Failure Information Used by GVS.

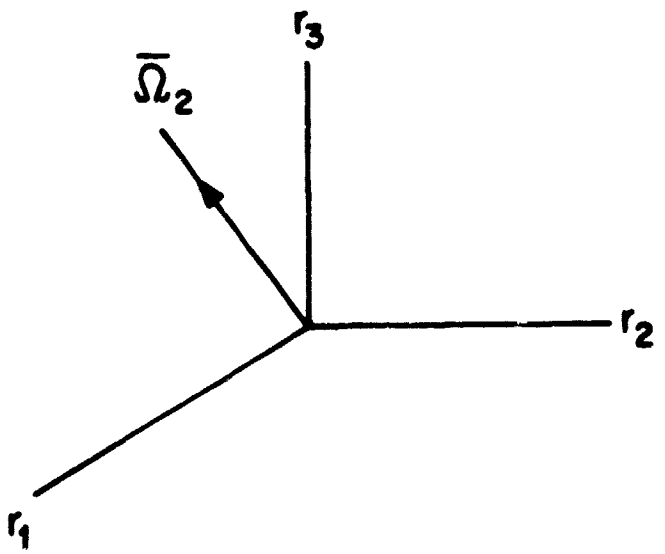


FIGURE 2-2b: Failure Information Used by the FDPS Method.

However, the failure effect is not necessarily confined to a fixed direction in parity space. To illustrate this, consider a residual vector r(k) based on the parity equations (2-37)-(2-39). We can write r(k) as

$$r(k) = \begin{bmatrix} 1 & -(a_{11}+a_{22}) & a_{11}a_{22} & 0 & 0 \\ 1 & -a_{12} & 0 & 0 & -a_{12} \\ 0 & 0 & 0 & 1 & -a_{22} \end{bmatrix} \begin{bmatrix} y_1(k) \\ y_1(k-1) \\ y_1(k-2) \\ y_2(k) \\ y_2(k-1) \end{bmatrix}$$

$$+ \begin{bmatrix} a_{22}-a_{12} & -1 & -(a_{22}+a_{12}) & 1 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_1(k-2) \\ u_2(k-1) \\ u_2(k-2) \end{bmatrix} \qquad (2\text{-}43)$$

When sensor 2 fails, its output is described by (2-41) and the residual becomes

$$r(k) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \nu(k) + \begin{bmatrix} 0 \\ -a_{12} \\ -a_{22} \end{bmatrix} \nu(k-1) \qquad (2\text{-}44)$$

Unless $\nu(k)$ is a constant function of time, the effect (signature) of a sensor 2 failure is only confined to a 2-dimensional subspace of the parity space. It is easy to see that this is also true for the other three components.

From the above observation, we can conclude that if a component is involved in more than one component of the residual (parity) vector and a window of more than one point in time of the signal associated with this component is used in generating the residual vector (i.e. if temporal redundancy is used), then the signature of this component failure is generally constrained to a multi-dimensional subspace of the residual space. Now each failure is associated with a subspace in the parity space. These subspaces in general may overlap with one another, or some may be contained in others. If no such subspace associated with a failure is contained in another, FDI can still be performed by determining which subspace the residual lies in. (We note that the detection filters of Beard [7] and Jones [8] effectively act in a closed-loop fashion to confine the signature of an actuator failure to a single direction and that of a sensor failure to a 2-dimensional subspace in the residual space. We will discuss this in the next section). From (2-43), it is easy to see that the subspaces associated with the four components (2 sensors and 2 actuators) are all 2-dimensional but no two of them are the same. Hence, FDI can be accomplished using the mechanism described above, whereas the GVS would not be applicable in this case (see Section 2.3).

The two forms of failure information discussed above represent time-independent failure characteristics, i.e. they are not dependent on detailed models of the (time history of the) failures as we have not assumed anything about the nature of $\nu(k)$. The temporal information buried in the residuals is also useful. It may be used to determine the nature of the failure or as added information to distinguish failures. For example, consider a sensor failure with signature described by (2-44). Under this failure, $r(k)$ generally

traverses a 2-dimensional subspace. If r(k) is observed to vary only along $[0 \quad -a_{12} \quad 1-a_{22}]'$, we can deduce that a constant bias has developed in this sensor. Turning this around, if we model the failure that we wish to look for as a bias, then we can look along the specific direction $[0 \quad -a_{12} \quad 1-a_{22}]'$. Also, if we model $\nu(k)$ in any other parametric form, e.g. as a ramp, we will specify a specific type of temporal trajectory for the signature. It is this type information that is used in the GLR algorithm, and it is this type of failure signature characterization that will form the starting point of our investigation of decision rules in Chapters 4,5, and 6.

To see how temporal information can help in distinguishing failures, let us consider r(k) given by (2-43). The subspaces associated with the two actuators are clearly not the same, but they overlap with each other. In a noisy environment it may be difficult to determine which subspace r(k) belongs to, especially when a major component of r(k) lies in the overlap. Now, suppose we have models for how the two failed actuators would behave over time. Then, we can determine the (temporal) signature of these failures (i.e. r(k) under each failure assuming no noise). Since these signatures describe the temporal behavior of the residual (the direction of r(k) for each k) under the corresponding failures, they represent more detailed characterization of failure information. As a result, distinguishability of these two failures can be improved by using a scheme that looks for these signatures in the residuals (by means of correlating the residuals with the signatures). Indeed, this is the basis for the GLR method.

From the above discussion we can see that in order to exploit temporal information, failure models are required. Moreover, decision processes making use of the temporal behavior of the residuals are more complex, because the failure signatures (now time functions) must be stored or generated on-line. Correlation of the residuals with the failure signatures will also add to the computational complexity.

In summary, we have described several forms of information contained in residuals generated from parity functions which are exploited for FDI. The simplest form is that employed by GVS. Failure directions (and subspaces) in parity space are time-independent failure characteristics utilized by several FDI schemes. Temporal information concerning failures, if available, can provide even more useful information for detecting failures, although there is a cost in additional system complexity.

2.5 FDI Systems Using Filters

In the previous section we discussed how the information contained in the residuals generated by parity functions is utilized for FDI. There is a large class of FDI systems such as GLR [4] and the detection filters of Beard [7] and Jones [8] that use filters to generate residuals for FDI. In these residual-generation processes, analytical redundancy is not used in the same explicit form as in the processes discussed in the last section. Here we will attempt to determine its relationship with the methods discussed in Section 2.4.

The general form of the filter we will consider is given by the following equations

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) \tag{2-45a}$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + Kr(k+1) \tag{2-45b}$$

$$r(k+1) = y(k+1) - C\hat{x}(k+1|k) \tag{2-45c}$$

where $\hat{x}(k|t)$ is the estimate of x at k given $y(1),\ldots,y(t)$, $t \leq k$; C is a matrix whose rows are the c_j's; y is the vector of the m sensor outputs; K is the filter gain that is chosen differently in different FDI schemes, r is the filter innovations and which are the residuals used for FDI. Since the prediction $\hat{x}(k+1|k)$ is based on the system dynamics (2-5) the filter has already made use of temporal redundancy of the system. Note that $C\hat{x}(k+1|k)$ is the prediction of $y(k+1)$ based on $y(1),\ldots,y(k)$, the residual given by (2-45c) is a vector analog of the closed-loop residual (r^3) discussed in Subsection 2.2.2. In contrast to an open-loop residual-generation process (whose main purpose is FDI), a filter actually serves two functions: to provide an estimate of the state in the normal mode (no failure) and to generate residuals for FDI.

In the absence of a failure, the innovations are given by

$$e(k+1) = \tilde{A}e(k) \tag{2-46}$$

$$r(k) = Ce(k) \tag{2-47}$$

where

$$e(k) = x(k) - \hat{x}(k|k-1) \tag{2-48}$$

$$\tilde{A} = A[I-KC] \tag{2-49}$$

$\tilde{A}$ is the closed-loop filter matrix, and e is the error of the state estimate. Under an actuator failure, the residual is characterized by

$$e(k+1) = \tilde{A}e(k) + fz(k) \tag{2-50}$$

and (2-47), where f is a vector (corresponding to b_j associated with the failed actuator) and z(k) is some scalar time function representing the temporal characteristics of the failure. When a sensor j failure occurs, the residual becomes

$$e(k+1) = \tilde{A}e(k+1) + (AK)_j \nu(k) \tag{2-51}$$

$$r(k) = Ce(k) + e_j\nu(k) \tag{2-52}$$

where $(AK)_j$ is the j-th column of the matrix product AK, e_j is the m-dimensional vector with the j-th element being one and all remaining elements equal to zero, and $\nu(k)$ is the scalar time function representing the temporal characteristics of the failed sensor. Therefore, a component failure affects the residuals through the matrices $\tilde{A}$ and C.

Note that $\tilde{A}$ depends on the filter gain K (2-49). As a result, the failure effect on r can be controlled to some extent by a choice of K. For different FDI schemes K is chosen to achieve different effects. For example, in GLR K is chosen so that (2-45) is a Kalman filter, i.e. r(k) is a zero mean white sequence under the no-fail condition. With this choice of K, failures generally do not produce special effects such as fixed directions in the residual space. For hypothesized z(k) or $\nu(k)$, the failure signatures, which are time functions, can be calculated. The GLR scheme achieves FDI

by recognizing such signatures contained in the residuals, and it makes heavy use of the temporal information.

The detection filter [7][8]* for an actuator failure has a special filter gain that makes $C\tilde{A}^i f$, i=0,1,...,n-1, lie in the same direction as Cf. Thus, this actuator failure produces a directional effect on r, and the detection of this failure can be accomplished by checking the residual for a component in this direction. For a sensor failure, the gain of the detection filter is chosen such that the failure signature is constrained to a 2-dimensional subspace of the residual space. (Because of the term $(AK)_j \nu(k)$, which actually depends on the gain, and $e_j \nu(k)$ in (2-51) and (2-52), it is generally only possible to choose a K so that a sensor failure signature is confined to a 2-dimensional subspace [8]). Therefore, the detection filters produce residuals carrying directional signatures, which are similar to those of the open-loop residuals discussed in the last section.

In summary, we note that residual-generation using filters is based on temporal redundancy. However, failure signatures are directly affected by the choice of the filter gain. In GLR the gain is chosen to whiten the residual, and the failure signatures are arbitrary time functions. Consequently the GLR schemes relies on temporal information for FDI. In a detection filter, directional failure signatures are produced in a closed-loop fashion via a proper choice of the filter gain, and the FDI mechanism is similar to that used with the open-loop residuals with directional signatures (see Section 2.4). Therefore, similar types of failure information may be produced via open-loop

* The original work of Beard and Jones concerns the continuous time problem, but the theory readily extends to the discrete time case where the system matrix $\tilde{A}$ is invertible. Also, it is possible to design a single detection filter for detecting several failures.

or closed-loop residual-generation process.

The discussion included in this section represents a preliminary effort in trying to understand how closed-loop FDI systems use analytical redundancy. Further research in the future is required for a thorough understanding of this subject.

CHAPTER 3

DESIGN OF ROBUST RESIDUAL GENERATION PROCESSES

3.1 Introduction

In Chapter 2, we presented a unified view of analytical redundancy in terms of generalized parity space, and we also discussed how parity functions can be directly used to generate residuals in the open-loop fashion for FDI. When the system model is exact and there is no noise disturbance, Chapter 2 provides the framework for obtaining the exact parity equations relating the various sensor outputs and actuator inputs. However, modelling uncertainties and noise effects will corrupt the parity relations. Thus, residual-generation process based on any nominal deterministic system model will be to some degree sensitive to modelling errors and noise. In this chapter, we will examine the problem of designing residual-generation processes that are robust in the presence of modelling uncertainties and noise disturbance.

Thoughout this chapter, we will assume that the structure of the system under consideration is known to have the form (2-5)-(2-6), and we are only uncertain of the exact values of some of the elements of the system matrices. That is, we have the following system model that includes both modelling uncertainties and noise disturbances,

$$x(k+1) = A(\gamma)x(k) + \sum_{j=1}^{q} b_j(\gamma)u_j(k) + \xi(k) \qquad (3\text{-}1)$$

$$y_j(k) = C_j(\gamma)x(k) + \eta_j(k), \qquad j=1,\ldots\ldots,m \qquad (3\text{-}2)$$

where γ is the vector of N uncertain parameters of the model, and we assume that $\gamma \in \Gamma$ where Γ is a known range of parameter values ($\gamma \in \Gamma \subset R^n$).

The dependence of A, b_j and c_j on γ indicates that their elements may be (different) functions of the parameter vector. This allows the modelling of elements in the system matrices as uncertain quantities that may be dependent on one another. The vectors ξ and $\eta = [\eta_1 \ldots \eta_m]$ are independent, zero-mean, white Gaussian noise vectors with constant covariance matrices $Q(\geq 0)$ and $R(> 0)$, respectively.

In this chapter we will concentrate on the problem of determining optimal parity functions, where "optimality" will be defined in terms of a measure of how large a deviation from zero could occur in a given parity relation in the presence of model uncertainties. Parity relations determined in this manner can be directly used in an open-loop FDI system, and our technique essentially provides the optimum design for this application. Of course, these parity relations can also be used to define ARMA models based on which closed-loop filters can be constructed to generate residuals. Although our design is aimed directly at optimizing the robustness of such a open-loop algorithm, one would generally expect that a parity relation that is robust open-loop could also be good for closed-loop residual-generation. The problem of determining optimal parity relations for closed-loop residual-generation should be investigated in the future, and our work provides the framework for such an investigation.

Before we proceed to describe the nature of the design problem at hand, it is useful to define the <u>structure</u> and the <u>coefficients</u> of a parity function. Recall that a parity function is a weighted combination of the actuator inputs and sensor outputs (see (2-18)). The structure of a parity relation refers to the set of input and output terms and the associated sets of time lags for each that are included in the parity function. For example, consider the

parity relations

$$y_1(k) + 7y_1(k-1) + 32y_2(k-1) = 0 \tag{3-3}$$

$$y_1(k) - .1y_1(k-1) + 5y_2(k-1) = 0 \tag{3-4}$$

$$y_1(k) + 3y_1(k-1) + 4y_2(k-1) - .01u_1(k-1) = 0 \tag{3-5}$$

Relations (3-3) and (3-4) have the same structure, because both parity functions include $y_1(k)$, $y_1(k-1)$ and $y_2(k-1)$; (3-3) and (3-5) have different structures, because (3-5) include the additional $u_1(k-1)$ term. The coefficients of a parity function refer to the (non-zero) coefficients of the sensor output and actuator terms in the parity function. For example, the parity coefficients of (3-5) are 1,3,4, and -.01.

A redundancy relation is specified by a parity structure and a set of parity coefficients. Any parity function (p) can be written in the form

$$p = \alpha\tilde{Y}(k) - \beta\tilde{U}(k) \tag{3-6}$$

where $\tilde{Y}(k)$ and $\tilde{U}(k)$ are vectors consisting of the sensor outputs and actuator inputs included in the parity function, respectively; the row vectors α and β represent the coefficients of the sensor output and actuator input terms in the parity function, respectively. Under the ideal conditions of a deterministic systems whose parameters are known exactly, α and β contain the non-zero elements of ω and $\omega\mathcal{B}$ of (2-12). For example, we have the following $\tilde{Y}(k)$, $\tilde{U}(k)$, α, and β for (3-5)

$$\tilde{Y}(k) = \begin{bmatrix} y_1(k-1) \\ y_1(k) \\ y_2(k-1) \end{bmatrix}$$

$$\tilde{U}(k) = [u_1(k-1)]$$

$$\alpha = [3 \quad 1 \quad 4]$$

$$\beta = [.01]$$

Therefore, in the above notation, the components of $\tilde{Y}(k)$ and $\tilde{U}(k)$ define the parity structure, and α and β represent the parity coefficients.

We now proceed to describe our approach to the design of robust residual-generation processes. This approach is best illustrated in terms of an example. Recall the three parity equations corresponding to the 2-dimensional example (2-30) considered in Section 2.3.

$$y_1(k)-(a_{11}+a_{22})y_1(k-1) + a_{11}a_{22}y_1(k-2) - a_{12}u(k-2) = 0 \qquad (3\text{-}7)$$

$$y_1(k) - a_{11}y_1(k-1) - a_{12}y_2(k-1) = 0 \qquad (3\text{-}8)$$

$$y_2(k) - a_{22}y_2(k-1) - u(k-1) = 0 \qquad (3\text{-}9)$$

In Section 2.3 we indicated that only two of the above parity relations need to be used in a generalized voting system to detect and identify a single component failure. When the elements of A, i.e. a_{11}, a_{12}, and a_{22} are perfectly known and there is no noise, all combination of two of the above parity functions will serve equally well for generating residuals.

When the exact values of a_{11}, a_{22}, and a_{22} are not known, (3-7)-(3-9) only specify the structures of three parity functions whose coefficients are uncertain quantities. In order to make use of any of these parity structures, we have to determine an appropriate set of coefficients for it. Ideally, the value of a parity function is zero in the absence of a failure. Under noise disturbances and modelling uncertainties, any choice of parity coefficients will result in a non-zero value for the parity function even when there is no failure. Hence, a natural design objective is to choose a set of parity coefficients that will make the parity function, in some sense, as close to zero as possible in the absence of a failure. Such a choice of coefficients effectively minimizes some measures of the effects of noise and modelling error on the parity functions. We will call this minimized measure of adverse effects the parity error (and we will define it precisely in Section 3.3).

When we have chosen the parity coefficients for all three parity structures, we will have also determined their corresponding parity errors. Then, it is clear that the residual-generation process should be based on the two parity functions that provide the largest failure signature to parity error ratios.

From the above discussion, it is evident that our design approach consists of three steps: 1) identify the useful parity structures, 2) determine the coefficients and the parity errors for these parity structures, and 3) determine the signature to parity error ratios that are used in deciding which parity functions are to be used for open-loop residual-generation. We will examine these three design steps in Sections 3.2, 3.3 and 3.4, respectively.

The parity coefficient design problem will be formulated as a minimax optimization (Section 3.3). The solution to such a problem is generally difficult. In Section 3.5, we will discuss a simple numerical example from which we can derive some useful insight into the solution procedure of the general minimax optimization. Finally, a summary of our approach to the design of robust residual-generation processes is included in Section 3.6.

3.2 Parity Structures Under Modelling Uncertainties

In this section, we will discuss how to obtain parity structures when there is uncertainty in the system model. First, we will review how a parity structure is obtained under the ideal condition (exact model and no noise). A parity function is determined from a set of linearly dependent rows of $C_j(\bar{n}_j+1)$, j=1,...m. Let C be the matrix composed of this set of dependent rows. Corresponding to these rows there are the components of $y_j(k,\bar{n}_j)$, j=1,...,m, which are collected together to form the vector $\tilde{Y}(k)$, and there are the rows of $\beta_j(\bar{n}_j)$, j=1,...,m, which are collected into the matrix $\tilde{\beta}$. Thus we can write a parity function as

$$p = \omega[\tilde{Y}(k)-\tilde{\beta}U(k,\bar{n}_0)] \tag{3-10}$$

with $\omega C=0$. Note that p is used in this chapter to denote a scalar parity function. Since not all components of $U(k,\bar{n}_0)$ are necessarily involved in a parity relation, we may collect all the non-zero columns of $\tilde{\beta}$ into a β and the corresponding components of $U(k,\bar{n}_0)$ into $\tilde{U}(k)$. Then we have

$$p = \omega\tilde{Y}(k) - \omega\beta\tilde{U}(k) \tag{3-11}$$

which is in the form of (3-6). The structure of this parity function is defined by the components of $\tilde{Y}(k)$ and $\tilde{U}(k)$. In terms of the notation defined above, we have

$$\tilde{Y}(k) = \mathcal{C}x(k) + \mathcal{B}\tilde{U}(k) \tag{3-12}$$

In the presence of modelling uncertainties and noise, (3-12) becomes

$$\tilde{Y}(k) = \mathcal{C}(\gamma)x(k,\gamma) + \Phi(\gamma)\tilde{\xi}(k) + \tilde{\eta}(k) + \mathcal{B}(\gamma)\tilde{U}(k) \tag{3-13}$$

where

$$\tilde{\xi} = [\xi(k) \ \ldots \ \xi(k+\rho)]' \tag{3-14}$$

and ρ is the order of the parity function. If the i-th component of $Y(k)$ is $y_s(k+\ell_1)$, then

$$\tilde{\eta}_i(k) = \eta_s(k+\ell_1) \tag{3-15}$$

and the i-th row of Φ, i.e. $\Phi(i)$, is

$$\Phi(i) = [c_s(\gamma)A^{\ell_1-1}(\gamma) \ \ldots \ c_s(\gamma)0\ldots0] \tag{3-16}$$

and Φ has ρ columns. It follows from (3-1) and (3-2) $\tilde{\xi}$ and $\tilde{\eta}$ are independent, zero-mean Gaussian random vectors with covariances

$$E\{\tilde{\xi}(k)\tilde{\xi}'(k)\} = \tilde{Q} = \begin{bmatrix} Q & & 0 \\ & \ddots & \\ 0 & & Q \end{bmatrix} \tag{3-17}$$

$$E\{\tilde{\eta}(k)\tilde{\eta}'(k)\} = \tilde{R} \tag{3-18}$$

and the (i,j)-th element of $\tilde{R}$ is

$$\tilde{R}_{ij} = R_{st}\delta_{\ell_1,\ell_2} \tag{3-19}$$

where δ_{ℓ_1,ℓ_2} is the Kronecker delta function, R_{st} is the (s,t)-th element of R, and we have assumed that $\tilde{Y}_j(k) = y_t(k+\ell_2)$. Note that $x(k,\gamma)$ is a random vector that is uncorrelated with $\tilde{\xi}(k)$ and $\tilde{\eta}(k)$ and

$$E\{x(k,\gamma)\} = x_0(k,\gamma) \tag{3-20}$$

$$E\{[x(k,\gamma) - x_0(k,\gamma)][x(k,\gamma) - x_0(k,\gamma)]'\} = \Sigma_x(\gamma) \tag{3-21}$$

where $\Sigma_x(\gamma)$ is the steady state covariance of $x(k,\gamma)$, which is dependent on γ through $A(\gamma)$ and $B(\gamma)$. In (3-13) we have also explicitly shown the dependence of C and $\bar{B}$ on the parameter γ.

Now we will consider parity structures under modelling uncertainties and noise effects. When (3-13) holds, the rows of $C(\gamma)$, which are chosen based on some nominal value γ_0 of γ, may not be linearly dependent for some other $\gamma\in\Gamma$. Even if they are linearly dependent for all $\gamma\in\Gamma$, for any choice of the parity coefficients, p (3-11) is generally not zero in the absence of a failure. This is because ω satisfying $\omega C(\gamma_0)=0$ does not generally imply $\omega C(\gamma)=0$ for all $\gamma\in\Gamma$. However, the parity structure of (3-11) is useful if we can find a set of parity coefficients that will make p <u>close</u> to zero under the no-fail condition. From this point of view, it is not necessary for a parity structure to be based on a C that is composed of linearly dependent rows of $C_j(\bar{n}_j+1)$, j=1,...,m, <u>as long as we can determine</u>

a set of appropriate coefficients that will result in a p that is close to zero when there is no failure. Then, the procedure of obtaining a parity structure described in the beginning of this section can also be applied to a C that is composed of some arbitrary set of rows of $C_j(\bar{n}_j+1)$, j=1,...,m. The usefulness of such a parity structure depends on the choice of coefficients which will be considered in the next section.

Using the above reasoning, we can obtain many candidate parity structure for residual-generation. However, not all of these structures are useful for a particular FDI system. Consider a generalized voting system, for instance. We need a set of parity functions that satisfy: 1) all components of interest must be included in at least one parity function, and 2) all components (possibly except one) must be excluded from at least one parity function. Requirements such as these help in limiting the number of candidate structures to be considered. In most applications, special feature of the system matrices will provide additional insights in the choice of parity structure. We will not address this problem further, but will focus on optimizing the set of parity coefficients once we have chosen a parity structure. Then we can use the results of this optimization to compare the usefulness of different parity structures.

3.3. Design of Parity Coefficients

Here, we will examine the problem of determining an appropriate set of coefficients (α,β) for a given parity structure (3-6)

$$p = \alpha\tilde{Y}(k) - \beta\tilde{U}(k) \qquad (3\text{-}22)$$

when we have noise and uncertainties in the system modelled by (3-1)(3-2). In the following, we will describe a formulation of this design task as a minimax optimization problem.

Consider the parity function (3-22). Under modelling uncertainties and noise, p is generally not zero for any choice of α and β. It is in fact a function of $\alpha,\beta,\gamma,x,\tilde{u},\tilde{\xi}$, and $\tilde{\eta}$. Substituting (3-13) in (3-22) we have

$$p(\alpha,\beta,\gamma,x(k,\gamma),\tilde{U}(k)) = \alpha[C(\gamma)x(k,\gamma)+\Phi(\gamma)\tilde{\xi}(k)+\tilde{\eta}(k)+B(\gamma)\tilde{U}(k)] - \beta\tilde{U}(k) \qquad (3\text{-}23)$$

Ideally, for (3-23) to be a parity function, p must be zero under the no-fail condition. Therefore, in order for p to be a useful parity function, we have tho choose α and β such that p is close to zero in the absence of a failure.

Due to the noises $\tilde{\xi}$, $\tilde{\eta}$, and the random state $x(k,\gamma)$, p is a random variable. A convenient quantity indicating the magnitude of p is $E\{p^2(\alpha,\beta,\gamma,x(k,\gamma),\tilde{U}(k)\}$, where the expectation is taken with respect to the joint probability density of $x(k,\gamma),\tilde{\xi}(k)$, and $\tilde{\eta}(k)$ (assuming the parameter vector has the value γ). Define

$$e(\alpha,\beta,x_0(k,\gamma^*),\Sigma_x(\gamma^*),\tilde{U}(k)) = \max_{\gamma\in\Gamma} E\{p^2(\alpha,\beta,\gamma,x(k,\gamma),\tilde{U}(k)\} \qquad (3\text{-}24)$$

where γ^* is the value of γ that solves the maximization, and $x_0(k,\gamma^*)$ and $\Sigma_x(\gamma^*)$ are the mean and covariance of $x(k,\gamma^*)$ (3-20), (3-21). The quantity e can be interpreted as the worst effect of the modelling error and noise on the parity function p. Then, we can attempt to achieve a conservative design by solving

$$\min_{\alpha,\beta} e(\alpha,\beta,x_0(k,\gamma^*),\Sigma_x(\gamma^*),\tilde{U}(k)) \tag{3-25}$$

As it stands the minimization problem (3-25) is not meaningful. Since p is linear in α and β, (3-25) has a trivial solution: $\alpha=0$, $\beta=0$. A parity function primarily relates the various sensor outputs, i.e. a parity function always includes sensor outputs but does not necessarily include actuator input (e.g. a direct redundancy relation). Therefore, α must not be zero. Without loss of generality, we can restrict α to be unit magnitude. The actuator input term in a parity function may be regarded as serving to make the parity function zero (compare (2-10) and (2-12)). Then, β is essentially free. However, we will now show that for each value of $\tilde{U}(k)$, β actually only has a single degree of freedom. For a given $\tilde{U}(k)$, any β can be written as

$$\beta = \lambda\tilde{U}'(k) + z \tag{3-26}$$

where λ is a scalar, and z' is a vector lying in the subspace orthogonal to $\tilde{U}(k)$, i.e. $z\tilde{U}(k)=0$. Hence, the component z in β will not produce any effect on p. This implies that we only have to consider β of the form

$$\beta = \lambda\tilde{U}'(k) \tag{3-27}$$

where λ represents the only degree of freedom of β.

Now we can construct a meaningful, optimization problem:

$$\min_{\alpha,\beta} \max_{\gamma\in\Gamma} E\{p^2(\alpha,\beta,\gamma,x(k,\gamma),\Sigma_x(\gamma),\tilde{U}(k))\} \tag{3-28}$$

$$\text{s.t.} \quad \alpha\alpha'=1$$

Using a β of the form (3-27) and (3-17), (3-18), and (3-23), we can write $E\{p^2\}$ as

$$E\{p^2(\alpha,\lambda,\gamma,x(k,\gamma),\tilde{U}(k))\}=[\alpha\lambda]S(\gamma,x_0(k,\gamma),\Sigma_x(\gamma),\tilde{U}(k))[\alpha\lambda]' \tag{3-29}$$

where S is the symmetric, positive-definite matrix for all γ

$$S(\gamma,x_0(k,\gamma),\tilde{U}(k)) = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \tag{3-30}$$

$$\begin{aligned} S_{11} &= C(\gamma)[x_0(k,\gamma)x_0'(k,\gamma)+\Sigma_x(\gamma)]C'(\gamma) + \Phi(\gamma)\tilde{Q}\Phi'(\gamma) + \tilde{R} \\ &+ B(\gamma)\tilde{U}(k)\tilde{U}'(k)B'(\gamma) + C(\gamma)x_0(k,\gamma)\tilde{U}'(k)B'(\gamma) \\ &+ B(\gamma)\tilde{U}(k)x_0'(k,\gamma)C'(\gamma) \end{aligned} \tag{3-31}$$

$$S_{12} = S_{21}' = -\mu[B(\gamma)\tilde{U}(k) + C'(\gamma)x_0(k,\gamma)] \tag{3-32}$$

$$S_{22} = \mu^2 \tag{3-33}$$

$$\mu = [\tilde{U}'(k)\tilde{U}(k)] \tag{3-34}$$

Note that S is dependent on $x_0(k,\gamma)$ and $\Sigma_x(\gamma)$, the mean and covariance of the system state, given γ is the true parameter vector. In general, $x_0(k,\gamma)$ and $\Sigma_x(\gamma)$ are very complex functions of γ. As a result, the maximization of $E\{p^2\}$ over γ is very difficult to perform. However, the problem may often be simplified by a reasonable approximation discussed in the following.

The principal feature of a parity function is to relate the outputs of different sensors and inputs to actuators at different points in time. The matrices $\mathbf{C}$, Φ, and B, which contain the dynamics of the system, represent the dominant feature of the parity relation. From this vantage point, the primary effect of the uncertainty in γ is manifested through $\mathbf{C}$, Φ, and B. Thus, it seems reasonable to approximate $x_0(k,\gamma)$ and $\Sigma_x(\gamma)$ by $x_0(k)$ and Σ_x corresponding to some nominal value of γ since the effect of γ on e through $x_0(k,\gamma)$ and $\Sigma_x(\gamma)$ is indirect and only of secondary importance. With this approximation, the dependence of S on γ is simplified, and the minimax problem takes the form

$$\min_{\alpha,\lambda} \max_{\gamma\in\Gamma} [\alpha\lambda] S(\gamma, x_0(k), \Sigma_x, \tilde{U}(k)) [\alpha\lambda]' \qquad (3\text{-}35)$$

$$\text{s.t. } \alpha\alpha' = 1$$

Despite the fact that the objective function of (3-34) is quadratic in α and λ, it is generally very difficult to solve, because S may depend on γ arbitrarily. In Section 3.5, we will discuss a simple example for which a solution procedure has been developed. There, we will also report some insights into the solution of the general problem obtained from this example.

We let α^* and λ^* denote the values of α and λ that solve (3-35), with $\beta^* = \lambda^* \tilde{U}(k)$, and

$$e^*(x_0(k), \Sigma_x, \tilde{U}(k)) = e(\alpha^*, \beta^*, x_0(k), \Sigma_x, \tilde{U}(k)) \qquad (3\text{-}36)$$

We call e^* the parity error of the parity function

$$p^* = \alpha^* Y(k) - \beta^* U(k) \tag{3-37}$$

Hence, e^* is a measure of the (minimized worst case) effect of modelling error and noise on (3-36). From the viewpoint of our design objective (3-28) e^* measures the fittness of (3-37) as a parity function.

As an aside, we note that α and β are not further constrained such that $\alpha C(\tilde{\gamma})=0$ and $\beta = \alpha B(\tilde{\gamma})$ for some value $\tilde{\gamma}\in\Gamma$. This is because even if they were, the resulting parity error would not be smaller than that of (3-36). In addition, just as for α^* and β^*, the α and β satisfying $\alpha C(\tilde{\gamma})=0$ $\beta = \alpha B(\tilde{\gamma})$ do not satisfy $\alpha C(\gamma_{true})=0$ and $\beta = \alpha B(\gamma_{true})$ in general. We now return to the minimax problem.

Note that the minimax solution α^* and $\lambda^*(\beta^*)$ are dependent on $x_0(k)$ and $\tilde{U}(k)$. In general, this means new coefficients would have to be computed at each time step if we wanted to continually achieve the optimum. This is clearly an undesirable requirement. Very often a set of coefficients will work well for a range of conditions, i.e. for x in some region of the state space. Therefore, a practical approach is to schedule the coefficients according to the operating condition. Each operating condition can be represented by a set-point, which is characterized by some nominal state x_0 and $\tilde{U}_0$ that are independent of time. Parity coefficients can be pre-computed (by solving (3-35) with x_0 and $\tilde{U}_0$ in place of $x_0(k)$ and $\tilde{U}(k)$) and stored. Then, the appropriate coefficients can be retrieved for use at the corresponding set-point. When the state and the inputs are varying slowly,

this scheme of scheduling coefficients is especially likely to deliver performance close to the optimum. In the remainder of this chapter we will focus on the design of parity coefficients for given set points.

The above formulation of the parity coefficient design problem may be slightly generalized to account for the fact that the true actuator input may not be $\tilde{U}_0$ exactly. To accomplish this we will let the actuator input term $\tilde{U}(k)$ be $\tilde{U}_0 + \delta U(k)$ in the minimax problem (3-28). The term $\tilde{U}_0$ is the set-point input (yielding the set-point state x_0) and $\delta U(k)$ is a random process that may be used to model two effects. In a true set-point operation, the set-point is often maintained by means of feedback through a compensator. Thus, the actuator input contains $\tilde{U}_0$ and $\delta U(k)$, which represents the variation in the input due to feedback. Based on the structure of the compensator, an expression for $\delta U(k)$ can be derived and subsequently used in (3-28). We note that $\delta U(k)$ in this case will be correlated with the state x. Using such an input in (3-28) will lead to additional terms in the S matrix that are the covariance of δU and cross covariance of δU and x. Since U(k) is not a fixed term, β will no longer be of a single-degree-of freedom but completely free. As a result we will have to consider the full vector β in the minimax problem instead of the scalar variable λ as in (3-29). However, the basic form of the design problem is not altered.

In the case where the set-point design is used in the parity coefficient scheduling scheme described earlier, $\delta U(k)$ may be used to model deviations from $\tilde{U}_0$ that are due to time-varying inputs used to change the state, such as in a manuevering vehicle. This will allow us, to some degree, to account for

the fact that the input is not necessarily fixed at $\tilde{U}_0$ in the minimax design. For simplicity, we may model $\delta U(k)$ here as a white Gaussian process with covariance Σ_u that is uncorrelated with the state (although more complex models can be used). Again, the inclusion of such a δU will not change the structure of the minimax problem, which will be the same as in the previous case.

In this section, the minimax design method was developed based on the assumption that γ is unknown. If a probability distribution over Γ may be obtained (or postulated), an alternative design formulation of the parity coefficient design problem is possible. Namely, instead of minimizing $\max_{\gamma \in \Gamma} E\{p^2\}$, we can consider minimizing $\tilde{E}\{p^2\}$, where $\tilde{E}$ denote expectation with respect to the joint distribution of $x(k)$, $\tilde{\eta}$, $\tilde{\xi}$, and γ. Then, we are looking at the averaged effect of γ on p. The design problem simply becomes a constrained quadratic minimization (it is essentially an eigenvalue-eigenvector problem), and it is simpler to solve. Detailed investigation of the merits of this approach is left for a future study.

3.4 Choice of Parity Functions for Open-Loop Residual-Generation

In the last section we presented a method for determining a set of parity coefficients for any given parity structure and given nominal set-point x_0, $\tilde{U}_0$ that is best in the sense that it minimizes the maximum mean square value of the parity function under the no-fail condition. After applying this method to the candidate parity structures, we have a set of candidate parity functions.

In this section, we will discuss the criteria for choosing an appropriate set among the candidate parity functions for use in open-loop residual-generation processes.

Associated with each candidate parity function (3-28) is a parity error e^* (3-35). Since p^* is linear in α^* and β^*, the magnitude squared of the combined vector of parity coefficients $[\alpha,\beta]$ scales the parity error. Therefore, the parity errors associated with the candidate parity functions can be compared if they are normalized. We define the <u>normalized parity error</u> $\bar{e}^*$

$$\bar{e}^* = e^*/||[\alpha^*,\beta^*]|| \tag{3-41}$$

where

$$||[\alpha^*,\beta^*]|| = \{[\alpha^*,\beta^*][\alpha^*,\beta^*]'\}^{1/2} \tag{3-42}$$
$$= 1 + \beta^*(\beta^*)'$$

and the <u>normalized parity coefficients</u>

$$\bar{\alpha}^* = \alpha^*/||[\alpha^*,\beta^*]|| \tag{3-43}$$

$$\bar{\beta}^* = \beta^*/||[\alpha^*,\beta^*]|| \tag{3-44}$$

Then, we can consider normalized parity functions:

$$\bar{p}^* = \bar{\alpha}^* \tilde{Y}(k) - \bar{\beta}^* \tilde{U}(k) \tag{3-45}$$

It is now clear that normalized parity functions that have the smallest normalized parity errors are preferred, because they are close to being ideal parity functions.

However, this is not the only issue that must be considered in designing robust FDI systems. Note that the usefulness of a parity function as a residual depends on how visible failure effects are in comparison to the inherrent parity error. To illustrate, consider an example in which we assume an additive failure and we let g denote the additive terms that appear in the normalized parity function under this failure. (Note that g is the signature of this failure and it is dependent on α^* and β^*.) We define the signature to parity error ratio π as

$$\pi = |g|/[\bar{e}^*]^{1/2} \tag{3-46}$$

Suppose we have to choose between two parity functions for the FDI of a particular failure. Then, we should choose the one that gives a bigger π. For instance, consider the parity structure (3-8)(3-9) of the example discussed in Section 3.1. Suppose we have determined the normalized parity coefficient for these structures, and we have the parity functions

$$p_1 = .778\ y_1(k) + .545\ y_1(k-1) + .311\ y_2(k-1)$$

$$p_2 = .592\ y_2(k) - .474\ y_2(k-1) + .652\ u(k-1)$$

If the failure we are interested in detecting is a bias of magnitude ν in sensor 2, the signature g_1 and g_2 corresponding to p_1 and p_2 are

$$g_1 = .311\nu$$

$$g_2 = .118\nu$$

and

$$\pi_1 = .311\nu/[\bar{e}_1^*]^{1/2}$$

$$\pi_2 = .118\nu/[\bar{e}_2^*]^{1/2}$$

where $\bar{e}_1^*$ and $\bar{e}_2^*$ are the normalized parity errors of p_1 and p_2. Therefore, for detecting a bias in sensor 2, we should choose $p_1 (p_2)$ if $\pi_1 > (<) \pi_2$, i.e. if $\bar{e}_1^* < (>) 6.92\ \bar{e}_2^*$.

In summary, parity functions with small normalized parity errors are usually preferred. In considering the detection and identification of particular failures, the signature to parity error ratios should be compared. The requirement for using π is that the nature of the failure will have to be given. In the above example, the failure of interest is simply a bias in the second sensor.

3.5 A Numerical Example

In this section, we consider the coefficient design problem for a simple example. We will develop a simple solution procedure for this problem, and we will also discuss the relation between this solution method and the solution of more general coefficient design problems.

The Numerical Example

The system under consideration is a 4-dimensional system operating at a set-point with two actuators and three sensors. The values of the system matrices are shown in Table 3-1. Except for two elements of the A matrix,

$$A = \begin{bmatrix} .5 & -.7 & .7 & 0 \\ 0 & .8 & \gamma_1 & 0 \\ -1 & 0 & 0 & .1 \\ 0 & 0 & \gamma_2 & .4 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$c_1 = [0 \quad 0 \quad 1 \quad 0]$$

$$c_2 = [0 \quad 1 \quad 0 \quad 0]$$

$$c_3 = [0 \quad 0 \quad 0 \quad 1]$$

$\gamma_1 \in [.02, \ .2]$ NOMINAL $\gamma_1 = .1$

$\gamma_2 \in [-.2, \ -.1]$ NOMINAL $\gamma_2 = -.15$

TABLE 3-1: System Parameters

all other parameters of the system are known exactly. These two uncertain elements are assumed to be two independent parameters denoted by γ_1 and γ_2. The ranges of these parameters are also shown in Table 3-1. The system is stable over the specified range of parameter value. In addition, there are plant and sensors noises, but we will describe them when we discuss the numerical results later in this section.

Suppose we want to design a generalized voting system for detecting a sensor failure. Three candidate parity structures are

$$p_1 = \alpha_1 \begin{bmatrix} y_2(k) \\ y_2(k+1) \\ y_1(k) \end{bmatrix} \tag{3-47}$$

$$p_2 = \alpha_2 \begin{bmatrix} y_2(k) \\ y_1(k) \\ y_1(k+1) \\ y_1(k+2) \end{bmatrix} \tag{3-48}$$

$$p_3 = \alpha_3 \begin{bmatrix} y_3(k) \\ y_3(k+1) \\ y_1(k) \end{bmatrix} \tag{3-49}$$

The corresponding Φ and C matrices are shown in Table 3-2. Note that each $C(\Phi)$ depends linearly on either γ_1 or γ_2 and that the rows of C_2 are not linearly dependent for <u>any</u> value of γ_2. The parity structures under consideration do not include any actuator inputs due to the fact that c_1B, c_2B, c_2AB, and c_3B are all zero. This will not cause any severe restriction in the following discussion, because the absence of the actuator inputs does not change the structure of the minimax problem (3-35). Assuming only a single sensor failure may occur, only p_3 plus either p_1 or p_2 need to be used for residual-generation (because both p_1 and p_2 include sensors 1 and 2).

<u>Solution Procedure</u>

Here, we will discuss the procedure for determining the coefficients for the above parity structure and also discuss several direct extensions suggested by this procedure. We will discuss its relation to the solution of the more general coefficient problem at the end of this section.

Since the three parity structures are of the same form (i.e. $p=\alpha Y(k)$), we can consider one generic problem characterized by C, α, Φ, etc. (without the index associated with the particular parity structure). We will let γ denote the scalar parameter that appears in C (since C is dependent on only one parameter in each of the three parity structures). The minimax problem is of the form

$$\min_{\alpha} \max_{\gamma\in[\gamma^1,\gamma^2]} \alpha\, S(\gamma)\alpha' \qquad (3\text{-}50)$$

$$\text{s.t} \quad \alpha\alpha'=1$$

$$C_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & .8 & \gamma_1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \Phi_1 = \begin{bmatrix} -0- \\ 0\ 1\ 0\ 0 \\ -0- \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ -.5 & -.7 & -.7+.17_2 & .04 \end{bmatrix} \qquad \Phi_2 = \begin{bmatrix} & & -0- & \\ 0\ 0\ 1\ 0 & & & 0\ 0\ 0\ 0 \\ & & & \\ -1\ 0\ 0\ .1 & & & 0\ 0\ 1\ 0 \\ & & -0- & \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & \gamma_2 & .4 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \Phi_3 = \begin{bmatrix} -0- \\ 0\ 0\ 0\ 1 \\ -0- \end{bmatrix}$$

TABLE 3-2: The C and Φ Matrices.

where the dependence of S on the set-point x_0 is suppressed, and γ^1 and γ^2 denote the minimum and maximum of the parameter variation. Substituting C and Φ of Table 3-2 into the expression for S in (3-33), we can easily see that S is quadratic in γ (because C and Φ are linear in γ), and S is positive definite for any value of γ. Then, $\alpha S(\gamma)\alpha'$ is convex in γ. In order to see this, we can write $\alpha S(\gamma)\alpha' = w_2\gamma^2 + w_1\gamma + w_0 > 0$, where w_2, w_1, w_0 are scalars dependent on α.) It follows that the maximum of $\alpha S(\gamma)\alpha'$ for any value of α occurs at either γ^1 or γ^2, and (3-50) becomes

$$\min_{\alpha} \max [f_1(\alpha), f_2(\alpha)] \qquad (3\text{-}51)$$

$$\text{s.t.} \quad \alpha\alpha' = 1$$

where

$$f_i(\alpha) = \alpha S(\gamma^i)\alpha', \qquad i=1,2 \qquad (3\text{-}52)$$

To gain some insights into the solution of (3-51), let us consider a 2-dimensional version of this problem, i.e. suppose that S is 2x2. In Figure 3-1. we have shown the ellipses $f_i(\alpha) = s_i$, i=1,2 in the α-plane. As s_i increases, the ellipses grow in size. Recall that α is constrained to be magnitude one, and this constraint takes the form of a unit circle in the α-plane. Along the unit circle we can trace the value of $f_i(\alpha)$, and this is shown in Figure 3-2. There are basically two cases. The first case is when the minimum of both f_1 and f_2 (along the unit circle) fall below the other function (see Figure 3-2a). It is easy to see that the solution of (3-51) has to be at one of the intersection points $\underline{\alpha}^1$ or $\underline{\alpha}^2$, i.e. $f_1(\underline{\alpha}^1) = f_2(\underline{\alpha}^1)$ and $f_1(\underline{\alpha}^2) = f_2(\underline{\alpha}^2)$. In Figure 3-2a, the minimax solution is

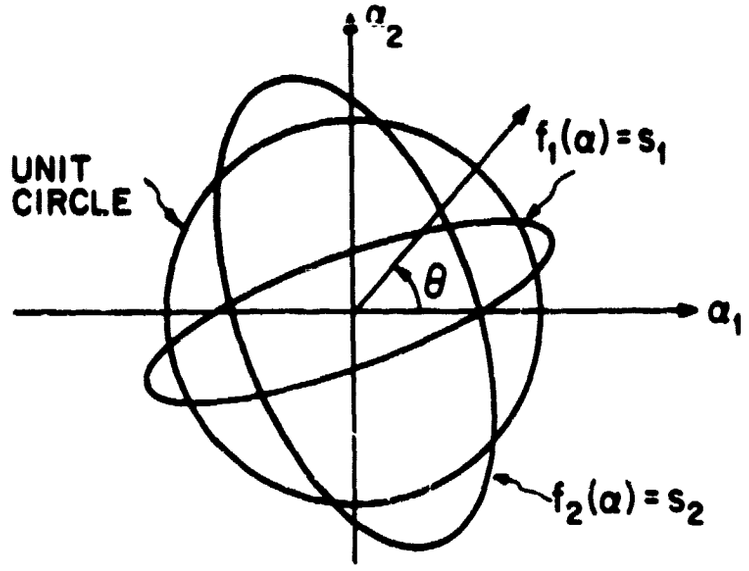


FIGURE 3-1: Ellipses in α-plane.

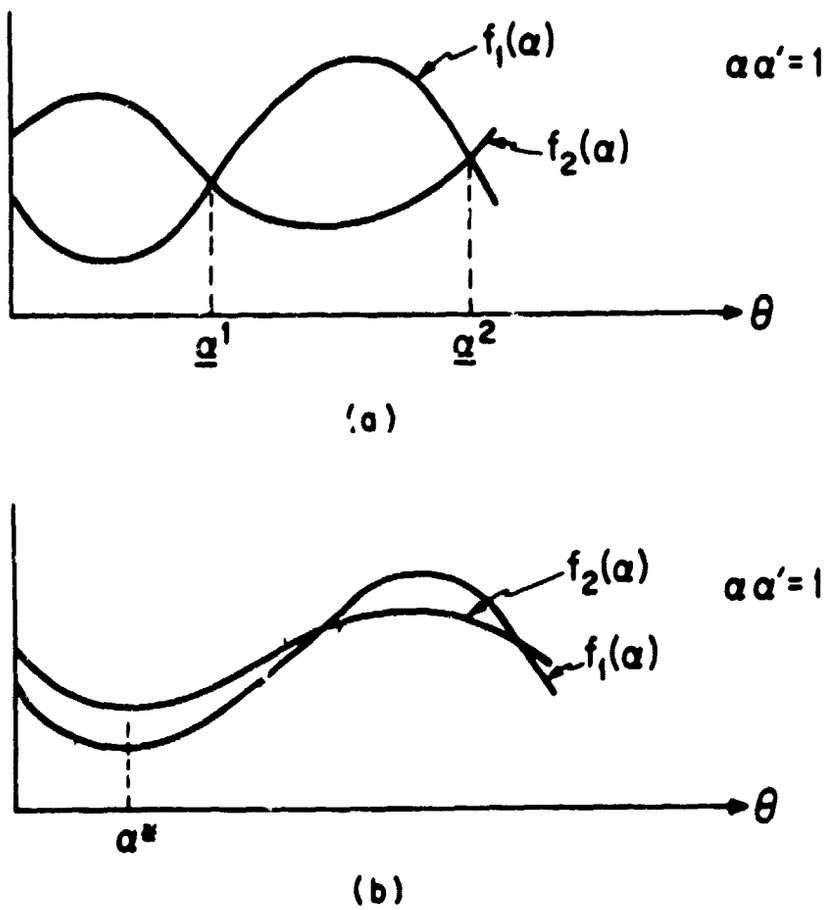


FIGURE 3-2: Value of f along circumference of unit circle in α-plane.

clearly at $\underline{\alpha}^1$, since $f_1(\underline{\alpha}^1) < f_1(\underline{\alpha}^2)$. The second case is when the minimum of either f_1 or f_2 remain above the other function (see Figure 3-26). Then it is clear that the minimum α^* is the minimax solution.

The above reasoning can be easily extended to higher dimensional problems, i.e. S is of arbitrary dimension, as long as $S(\gamma)$ is quadratic in γ. In fact, it can be easily shown that the solution of (3-51) can be obtained by a two-step procedure (see Appendix A). To simplify the description of this procedure, we need to define $\hat{\alpha}^1$, $\hat{\alpha}^2$ and Λ in following manner

$$f_i(\hat{\alpha}^i) = \min_{\alpha} \; f_i(\alpha) \qquad i=1,2 \qquad \text{s.t.} \quad \alpha\alpha'=1 \tag{3-53}$$

$$\Lambda = \{\alpha: f_1(\alpha) = f_2(\alpha),\ \alpha\alpha'=1\} \tag{3-54}$$

Since $f_1(\alpha) = \alpha S(\gamma^i)\alpha$ (S is symmetric and positive-definite), $\hat{\alpha}^i$ is simply the eigenvector of $S(\gamma^i)$ corresponding to the smallest eigenvalue.

The two-step solution procedure for (3-51) is:

1) The first step involves checking the conditions

$$f_1(\hat{\alpha}^1) \geq f_2(\hat{\alpha}^1) \tag{3-55a}$$

$$f_2(\hat{\alpha}^2) \geq f_1(\hat{\alpha}^2) \tag{3-55b}$$

If both $\hat{\alpha}^1$ and $\hat{\alpha}^2$ satisfy (3-55), then the $\hat{\alpha}^i$ giving a smaller $f_i(\hat{\alpha}^i)$ is the solution of (3-51). If only one $\hat{\alpha}^i$ satisfies (5-55), it is the solution.

If neither satisfies (5-55), then we have to perform step 2.

2) Search over the set Λ (3-54) to find an α that minimizes $f_1(\alpha)$ or $f_2(\alpha)$ (since over Λ, $f_1(\alpha) = f_2(\alpha)$). This is equivalent to solving a quadratic programming problem with quadratic constraints:

$$\begin{aligned} &\min_{\alpha} \quad f_1(\alpha) \\ &\text{s.t.} \quad \alpha\alpha'=1 \\ &\qquad \alpha[S(\gamma^1)-S(\gamma^2)]\alpha'=0 \end{aligned} \tag{3-56}$$

The solution of (3-56) is the minimax solution to (3-51), if step 1 fails to produces a solution. The minimization may be solved numerically using existing optimization techniques [14].

The above solution procedure can be readily extended to the case where S is a quadratic function of an N-vector γ and each element of γ is independently bounded by an interval (i.e. Γ is a hyper-rectangle). Then the minimax problem (3-51) becomes

$$\begin{aligned} &\min_{\alpha} \max_{i=1,\ldots,2^N} f_i(\alpha) \\ &\text{s.t.} \quad \alpha\alpha'=1 \end{aligned} \tag{3-57}$$

where $f_i(\alpha) = \alpha S(\gamma^i)\alpha'$, and γ^i, $i=1,\ldots,2^N$ denote the 2^N corners of Γ. The above solution procedure is modified as follows. In Step 1, we have to consider the minimax of all $f_i(\alpha)$. That is, we have to replace (3-55) by

$$f_i(\hat{\alpha}^i) = \min_{j=1,\ldots,2^N} f_j(\hat{\alpha}^i), \quad i=1,\ldots,2^N \tag{3-58}$$

In step 2, we will have to solve (3-56) for all combinations of $f_i(\alpha)$ and $f_j(\alpha)$, $i \neq j$. Then the minimum of all the solutions of the quadratic programs is the minimax solution.

This solution method can be extended to handle parity structures with actuator inputs. Of course, the corresponding S matrix has to be quadratic in γ for our approach to be valid. In Appendix A, we will discuss such an extension to include actuator inputs for the scalar case. The N-parameter case with input terms can be treated in the same way as the N-parameter case with no input term was treated in the above discussion.

Numerical Results

The minimax design problem has been solved for a set of six test conditions consisting of different set-points and different plant and sensor noise intensities. These test conditions are described in Tables 3-3 and 3-4. The two set-points [-4.16 7.03 4.06 -1.01]' and [4.06 2.90 5.80 -1.45]' can be obtained by applying $u_1=1$ and $u_2=10$ to the nominal system model. The nominal state covariances Σ_x^1 and Σ_x^2 due to the two different hypothesized plant noise intensities Q^1 and Q^2 are listed in Table 3-5.

A computer program based on the penalty-multiplier method [14] for solving non-linear optimization problems is used when it is necessary to carry out step two of this solution procedure. In addition to the symmetry (i.e. if α is a candidate solution of (5-56), so is $-\alpha$), there are generally local minima for (5-56). In order to obtain the global minimum, a large number of initial guesses may have to be used in the minimization program. For our

$$Q^1 = \begin{bmatrix} .25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .25 \end{bmatrix} \qquad Q^2 = \begin{bmatrix} .25 & 0 & -.325 & 0 \\ 0 & .5 & 0 & 0 \\ -.325 & 0 & .625 & 0 \\ 0 & 0 & 0 & .25 \end{bmatrix}$$

$$R = \begin{bmatrix} R_1 & & 0 \\ & R_2 & \\ 0 & & R_3 \end{bmatrix} \qquad R_i = 1,2, \quad i=1,2,3$$

TABLE 3-3: Noise Variances

COND. CODE	PARAMETERS
a	x_0 = [0 0 0 0] Q^1, DIAG R = [1 1 1]
b	x_0 = [-4.16 7.03 4.06 -1.01] Q^1, DIAG R = [1 1 1]
c	x_0 = [4.06 2.90 5.80 -1.45] Q^1, DIAG R = [1 1 1]
d	x_0 = [4.06 2.90 5.80 -1.45] Q^1, DIAG R = [1 2 2]
e	x_0 = [4.06 2.90 5.80 -1.45] Q^1, DIAG R = [2 1 1]
f	x_0 = [4.06 2.90 5.8 -1.45] Q^2, DIAG R = [1 1 1]

TABLE 3-4: Test Conditions.

$$\overset{1}{\Sigma}_x = \begin{bmatrix} .5580 & .0342 & -.1508 & -.0552 \\ .0342 & .0102 & -.0129 & -.0097 \\ -.1508 & -.0129 & .5772 & .0117 \\ -.0552 & -.0097 & .0117 & .3113 \end{bmatrix}$$

$$\overset{2}{\Sigma}_x = \begin{bmatrix} 1.958 & -.8434 & -1.114 & -.1049 \\ -.8434 & 1.803 & .7691 & -.1996 \\ -1.114 & .7691 & 2.608 & -.1081 \\ -.104 & -.1996 & -.1081 & .3829 \end{bmatrix}$$

TABLE 3-5: Nominal Σ_x

problems, we always used either $\hat{\alpha}^1$ or $\hat{\alpha}^2$ (corresponding to the minima of f_1 and f_2) as the initial guess for the second step, and the algorithm has always converged to an e^* quite rapidly. Other initial guesses were also tried, but they either led to the same solution or higher e values. Therefore, we may conclude that we have determined the global minima for our problems. The resulting coefficients are tabulated in Table 3-6.

For this example, it is evident that the parity coefficients are generally strongly dependent on the test condition (the values of x_0, Q, and R). Although this dependence is very complex, some insights may still be obtained from the numerical results. Consider, for instance, p_1 under conditions b and c. From Table 3-6, we have for condition b the parity function

$$\bar{p}^*_{1b} = .6411\ y_2(k) - .7666\ y_2(k+1) + .0378\ y_1(k) \tag{3-59}$$

and for condition c

$$\bar{p}^*_{1c} = .8947\ y_2(k) - .3667\ y_2(k+1) - .2551\ y_1(k) \tag{3-60}$$

The only difference between these two test conditions lies in the value of x_0 (see Table 3-5). Since the first and fourth column of C_1 (Table 3-2) are zero, only the second and third elements of x_0 (x_{02} and x_{03}) will play a role in the coefficient optimization problem. The parity function p_1 can be written in the form of (3-26):

$$p_1 = \alpha_1 x_{02} + \alpha_2 (x_{02} + \gamma_1 x_{03}) + \alpha_3 x_{03} + \zeta(\gamma_1, \alpha) \tag{3-61}$$

TEST COND.	PAR. FUNC.	$\overrightarrow{e}^*$	$\overline{\alpha}^*$			
	1	1.002	.7282	-.6808	.0791	
a	2	1.008*	.9983	.0223	.0483	.0219
	3	1.118*	.6833	-.7208	-.1167	
	1	1.082	.6411	-.7666	.0378	
b	2	1.101*	.4462	.5079	-.4356	.5942
	3	1.210	.7027	-.7115	-.064	
	1	1.096	.8947	-.3667	-.2551	
c	2	1.055	.9599	-.1484	.1992	.1296
	3	1.230	.7592	-.6504	.0249	
	1	1.908	.7865	.3023	-.5385	
d	2	1.123	.7345	-.5931	.4697	-.6559
	3	2.228	.7981	-.6007	.0684	
	1	1.124	.8058	-.5832	-.1025	
e	2	1.122*	.9669	-.1204	.1242	.1875
	3	1.230	.7441	-.6678	.1692	
	1	1.427	.7327	-.6803	-.0166	
f	2	1.311*	.5146	.4404	-.3312	.6570
	3	1.254	.6385	-.7687	.0375	

*Solution obtained in Step 1 of solution procedure

TABLE 3-6: Minimax Parity Coefficients and Parity Errors

where α_1, α_2 and α_3 denote the elements of α corresponding to $y_2(k)$, $y_2(k+1)$, and $y_1(k)$ respectively, and ζ denotes the remaining noise terms. It is clear that x_{03} and α_2 modulates the effect of γ_1 on p_1. Qualitatively, as $|x_{03}|$ becomes large relative to $|x_{02}|$ (with all the noise covariances the same), the optimal α_2 will reduce in size (relative to α_1 and α_3) in order to keep the effect of γ_1 small. As $|x_{03}|$ increases, the signal to noise ratio of $y_1(k)$ also increases. Therefore, we expect $|\alpha_3|$ to become large so as to make better use of the information provided by $y_1(k)$. Under conditions b, $x_{02}(=7.03) > x_{03}(=4.06)$, and under condition c, $x_{02}(=2.9) < x_{03}(=5.8)$. An inspection of (3-59) and (3-60) shows that the above reasoning holds.

Note that both p_1 and p_2 relate the first sensor with the second one, and p_2 is a higher order relation than p_1. Furthermore, the rows of C_2 are not linearly dependent for any value of γ_2. However, the parity error associated with p_2 is smaller than that of p_1 in all cases except condition a. This shows that a higher order parity function (which is likely to contain high order effects of γ_1) is not necessarily more vulnerable to modelling errors and noise. In addition, a parity function based on a C matrix with rows that are linearly dependent for all values of γ does not necessarily produce a smaller parity error than a parity function that is based on a C with independent rows.

In Table 3-7 we have tabulated the signature to parity error ratio (π_i) associated with the parity function for sensor bias failures under conditions c and d. (π_i denotes the signature to parity error ratio associated with a

TEST COND.	PARITY FUNCTION	π_1	π_2	π_3
c	p_1	$.243\nu_1$	$.504\nu_2$	-
	p_2	$.176\nu_1$	$.934\nu_2$	-
	p_3	$.022\nu_1$	-	$.107\nu_3$
d	p_1	$.390\nu_1$	$.788\nu_2$	-
	p_2	$.733\nu_1$	$.693\nu_2$	-
	p_3	$.046\nu_1$	-	$.126\nu_3$

TABLE 3-7: π Values for Selected Test Conditions

parity function for a sensor i bias failure of magnitude ν.) The parity function p_2 has a larger π_2 in condition c and a larger π_1 in condition d; p_1 has a larger π_1 in condition c. The parity function p_3 has very small π_1 and π_3 in both conditions c and d. Therefore, p_3 is not very useful for detecting a small bias in either sensor 1 or sensor 3.

Further Discussion of Minimax Solution

Earlier in this section, we discussed a simple method of solving the minimax problem when the objective function $\alpha S(\gamma)\alpha'$ is quadratic in γ. The simplicity of the solution is a direct consequence of the fact that the maximization of $\alpha S(\gamma)\alpha'$ with respect to γ (3-50) can be replaced by the maximum among $\alpha S(\gamma^i)\alpha'$ at the finite number of corners (γ^i) of Γ (which is assumed to be a hyper-rectangle) for all α. Whenever this replacement can be done, this simple solution method applies. This is possible, for instance, when $\alpha S(\gamma)\alpha'$ is convex over Γ for all α ($\alpha\alpha'=1$). Nonlinear dependence of the elements of A, B, and C on γ and higher order effects (due to products such as CA, CA^2, etc.) in $\bar{C}$, $\bar{\Phi}$, and $\bar{B}$ will make the objective function of the minimax problem ($\alpha S(\gamma)\alpha'$ or $[\alpha\lambda]S(\gamma)[\alpha\lambda]'$ when there are actuator terms) non-quadratic in γ. In such cases, the task of checking for convexity is generally difficult. Moreover, convexity is not a necessary condition for replacing the maximization of $\alpha S(\gamma)\alpha'$ over γ by the maximum of $\alpha S(\gamma^i)\alpha'$, i=1,...,M (where M is the number of corners of Γ). By examining the geometry of the problem, some insights have been obtained, and we will discuss them in the following.

Suppose there is only a scalar parameter γ and there are only two elements σ_1 and σ_2, among all elements of C and Φ, that are dependent on γ. (Although we will only consider $\alpha S(\gamma)\alpha'$ in this discussion, it is clear that the extension to include actuator terms is immediate.) Then, σ_1 and σ_2 are also dependent on each other. Let us write $h_\alpha(\sigma_1,\sigma_2) = \alpha S(\gamma)\alpha'$; $h_\alpha(\sigma_1,\sigma_2)$ is a positive quadratic function of σ_1 and σ_2 for any value of α. For each value of α, we can draw the ellipsis $h_\alpha(\sigma_1,\sigma_2)=\theta$ for different values of θ in the σ_1-σ_2 plane (Figure 3-3a). Note that the ellipsis increases in size with increasing θ.

Because σ_1 and σ_2 both depend on γ, they can only take on values along the curve $(\sigma_1(\gamma),\sigma_2(\gamma))$ with γ varying over its range Γ. The curve is characterized by a scalar equation, say $G_1(\sigma_1,\sigma_2)=0$ (see Figure 3-36). By tracing along this curve (called G_1) and noting the values of h_α and γ along it, we can obtain the function $\tilde{h}_\alpha(\gamma) = \alpha S(\gamma)\alpha'$ for a fixed α (see Figure 3-3b). It is evident that the maximum of $\tilde{h}_\alpha$ does not occur at either end of Γ (in the case shown in Figure 3-36). Now suppose the relationship between σ_1 and σ_2 is characterized by another curve $G_2(\sigma_1,\sigma_2)=0$ (Figure 3-4a). The plot of $\tilde{h}_\alpha$ versus γ for G_2 (Figure 3-4b) shows that the maximum of $\tilde{h}_\alpha$ is at γ^2.

Based on the geometry described above, we can re-state the condition under which $\max_\gamma \tilde{h}_\alpha(\gamma) = \max[\tilde{h}_\alpha(\gamma^1),\tilde{h}_\alpha(\gamma^2)]$ for a fixed α as follows. (We will let $\sigma = [\sigma_1,\sigma_2]'$ to simplify the notation). For each α, consider the ellipses $h_\alpha(\sigma) = \tilde{h}_\alpha(\gamma^1)$ and $h_\alpha(\sigma) = \tilde{h}_\alpha(\gamma^2)$ that pass through $\sigma(\gamma^1)$ and $\sigma(\gamma^2)$ respectively. Define γ^* such that

$$\gamma^* = \arg \max_{\gamma^1,\gamma^2} [\tilde{h}_\alpha(\gamma^1),\tilde{h}_\alpha(\gamma^2)] \qquad (3\text{-}6?)$$

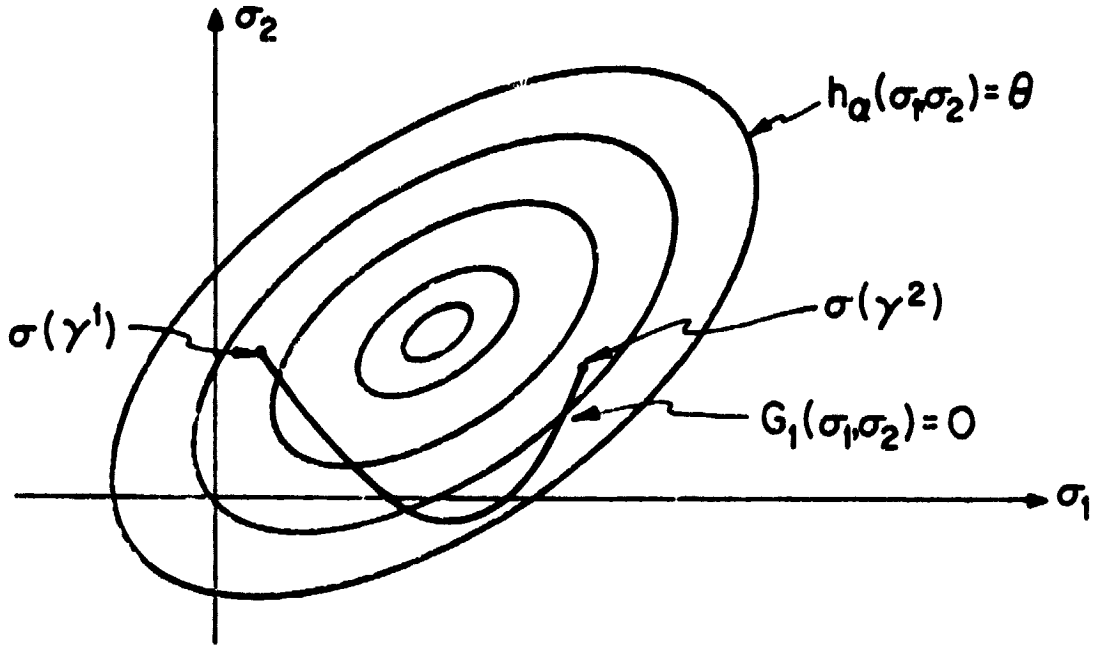


FIGURE 3-3a: Contours of $h_\alpha(\sigma_1\sigma_2)$ and G_1 for a fixed α.

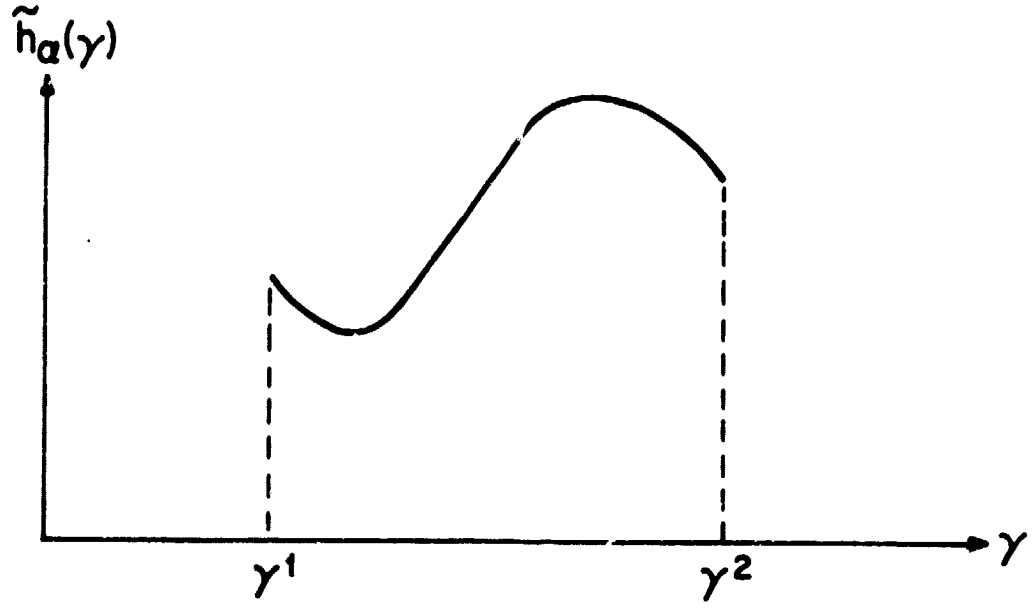


FIGURE 3-3b: $\tilde{h}_\alpha(\gamma)$ along G_1.

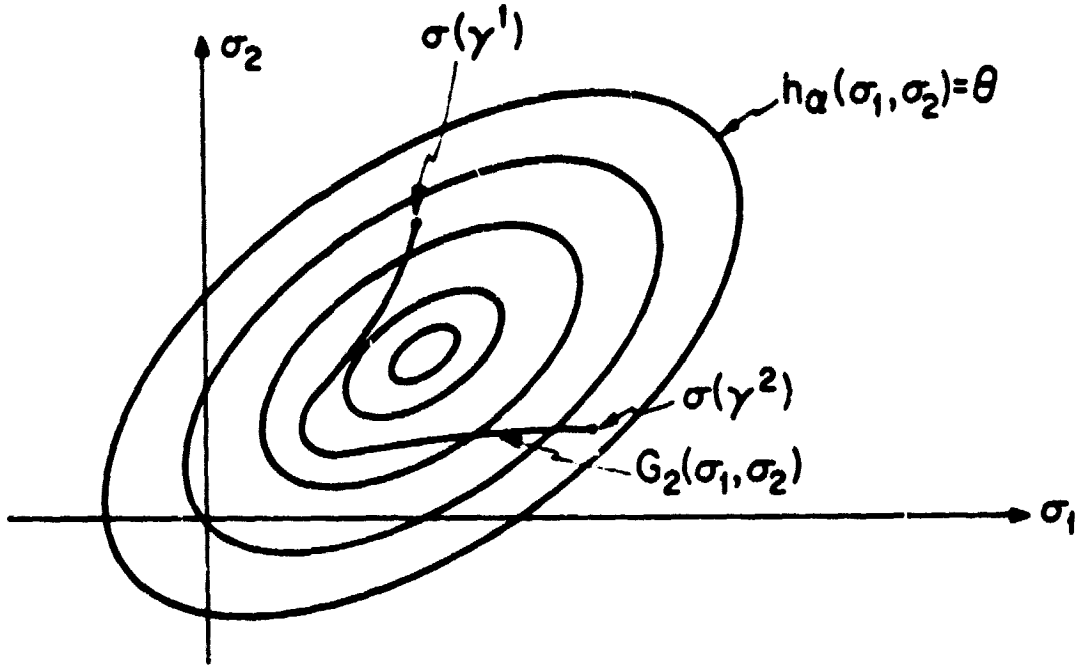


FIGURE 3-4a: Contours of $h_\alpha(\sigma_1\sigma_2)$ and G_2 for a fixed α.

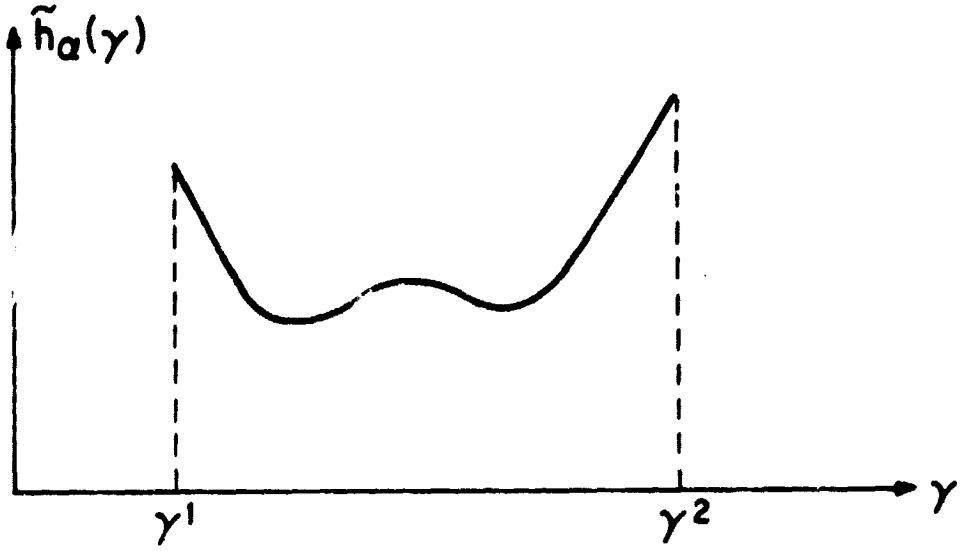


FIGURE 3-4b: $\tilde{h}_\alpha(\gamma)$ along G_2.

Note that γ^* is dependent on α, and $h_\alpha(\sigma) = \tilde{h}_\alpha(\gamma^*)$ is the bigger of the two ellipses passing through $\sigma(\gamma^1)$ and $\alpha(\gamma^2)$. Then, $\max_\gamma \tilde{h}_\alpha(\gamma) = \max[\tilde{h}_\alpha(\gamma^1), \tilde{h}_\alpha(\gamma^2)]$ if and only if the curve G (describing the relationship between σ_1 and σ_2) lies within the bigger ellipse $h_\alpha(\sigma) = \tilde{h}_\alpha(\gamma^*)$. Mathematically, this is equivalent to the condition

$$h_\alpha(\sigma) \leq \tilde{h}_\alpha(\gamma^*), \qquad \sigma \in \{\sigma: G(\sigma)=0\} \tag{3-63}$$

It should be noted that a brute-force approach to testing condition (3-63) will result in the evaluation of $\tilde{h}_\alpha(\gamma)$ for all $\gamma \in \Gamma$. A conceptually simpler approach to testing this condition is described below.

Consider the simultaneous equations

$$h_\alpha(\sigma) = \tilde{h}_\alpha(\gamma^*) \tag{3-64a}$$

$$G(\sigma) = 0 \tag{3-64b}$$

Assuming σ is continuous in γ (hence the curve G is continuous), we can deduce that if the set of solutions to (3-64) does not contain any point σ other than $\sigma(\gamma^1)$ and/or $\sigma(\gamma^2)$, then the curve G is either completely inside or outside the ellipse $h_\alpha(\sigma) = \tilde{h}_\alpha(\gamma^*)$ (and touching it only at $\sigma(\gamma^1)$ and/or $\sigma(\gamma^2)$). Then, the curve G lies inside the ellipse if the following is true.

$$h_\alpha(\bar{\sigma}) < \tilde{h}_\alpha(\gamma^*), \qquad \bar{\sigma} \in \{\sigma: G(\sigma)=0\} \tag{3-65}$$

Therefore, the testing of the condition (3-63) requires studying the solutions of the simultaneous equations (3-64). If the solution set of

(3-64) does not contain any σ other than $\sigma(\gamma^1)$ and $\sigma(\gamma^2)$ and (3-65) holds, then $\max_{\gamma} \tilde{h}_\alpha(\gamma) = \max[\tilde{h}_\alpha(\gamma^1), \tilde{h}_\alpha(\gamma^2)]$ for a fixed α.

In order for the simple minimax solution described earlier to apply, condition (3-63) has to hold for all α such that $\alpha\alpha'=1$. This implies that we have to examine the solution of (3-64) and condition (3-65) as a function of α. This is generally a difficult task, because, even for a fixed α, the solution of (3-64) is difficult to determine for an arbitrary G. Nevertheless, this approach provides an important perspective on the problem, i.e. the objective function $\alpha S(\gamma)\alpha$ is now separated into $h_\alpha(\sigma)$, which is independent of γ, and $G(\sigma)$, which contains all the effects of γ. This explicit isolation of the effects of γ will allow us to exploit the structure of G (i.e. the inter-dependence of σ_1 and σ_2 through this mutual dependence on γ) to determine if (3-63) holds. However, future work is required to develop this concept into a practical procedure for testing (3-63). In closing, we note that the above discussion for the case of 2 σ's and a scalar γ can be readily generalized to include multiple σ's and a vector γ.

3.6 Summary

In this chapter we have developed an approach to the design of robust residual-generation processes. We have examined in detail the three basic steps of the design method: i) the choice of parity structures in the presence of modelling uncertainties, ii) the design of parity coefficients, iii) the choice of parity functions for generating residuals based on signature to parity error ratios.

C-2

The conceptualization of analytical redundancy as parity relations together with the design method described in the chapter represent a new approach to the design of residual-generation processes for FDI. The formulation of the parity coefficient design problem as a minimax optimization provides a basis for exploitating analytical redundancy in a robust manner. Although the minimax problem is difficult to solve, a simple solution procedure has been found for some special cases. The insights provided by this solution method will aid in the study of the solution if more general minimax problems required to design robust residual-generation schemes.

CHAPTER 4

OPTIMAL SEQUENTIAL DECISION RULES

4.1 The Sequential Decision Problem-Background

The output of the residual-generation process is the random residual sequence, r(k). The behavior of the residual is described by a set of probability density functions $\{p(r(1),\ldots,r(k)|(i,\tau,\nu)),\ k=1,2,\ldots,\}$ that is characteristic of the presence of the failure (i,τ,ν) (the notation $(i,\tau,\nu)=(0,-,-)$ denotes the absence of any failure), and such probability density functions represent the signature of the failure. The FDI process involves monitoring the residual for changes from its normal (no-fail) behavior. Residual samples are observed in sequence. If a failure is judged to have occurred and sufficient information (from the residual) has been gathered, the monitoring process is stopped. Then, based on the past observations of residual, an identification of the failure is made. If no failure has occurred, or if the information gathered is insufficient, monitoring is not interrupted so that further residual samples may be observed. The decision to interrupt the residual-monitoring to make a failure identification is based on a compromise between the speed and accuracy of the detection, and the failure identification reflects the design tradeoff among the errors in failure classification. Such a decision mechanism belongs to the extensively studied class of sequential tests or sequential decision rules. In this research, we will extend existing concepts and formulations of the sequential decision problem to the design of decision rules for FDI systems.

The first important piece of work in sequential analysis was presented in 1947 by Wald [10], where the Sequential Probability Ratio Test (SPRT) was proposed as a procedure for testing a simple hypothesis against a simple

alternative. It was shown, a year later, by Wald and Wolfowitz [18] that the SPRT is optimal in the sense that among the class of sequential tests that have misclassification errors not greater than those of the SPRT, the SPRT will take the smallest expected number of samples to reach a decision. The conceptual and structural simplicity of the SPRT has made it the basis of many studies in the design and application of sequential tests. For example, the SPRT was employed as a means of identifying aircraft sensor failures [11], and SPRT- like tests were developed for robust signal detection [19]. In addition, modifications such as those investigated by Anderson [20] and Chien and Adams [21] have been suggested to enable the SPRT to deal with a wider class of problems in a satisfactory manner. Specifically, Anderson has modified the decision thresholds of the SPRT so as to maintain a reasonable expected sample size when the same SPRT is used in testing a simple hypothesis against a composite alternative, and Chien and Adams have introduced resets for the SPRT in order to detect a change from one hypothesis to another at some unknown time. In the latter case, the change in the hypothesis at an unknown time indeed models the occurrence of a (the only possible) failure.

The decision problem to which the SPRT is the solution is a special and very simple case of the general Bayes Sequential Decision Problem (BSDP) first studied by Arrow, Blackwell and Girshick and later described by Blackwell and Girshick [22]. The BSDP provides a general formulation that can be adapted to many meaningful decision problems. In particular, it is suitable for the FDI decision problem, where the occurrence of a failure may be regarded as a change from the normal (no-fail) hypothesis to some failure hypothesis at an unknown time. The optimal Bayes Sequential Decision Rule (BSDR) has a form

similar to that of the dynamic programming solution to the discrete time optimal control problems [17]. Even with the aid of modern computers, the BSDR can rarely be calculated for the general problem, and only in some limiting cases [23] [24] has some feel for the solution been obtained. For this reason, alternative suboptimal sequential procedures were often investigated as means of solving sequential decision problems e.g., [25] [26] [27].

The inclusion of changes among hypotheses at unknown times further complicates the computational aspect of the BSDR. As noted by Chernoff and Zacks in their study of estimating a parameter which may change in time [28], only suboptimal or ad hoc procedures are practical solutions. However, for the case where only a change from a simple hypothesis to a simple alternative is allowed, some useful results are available. As mentioned earlier, Chien and Adams were able to modify the SPRT to accommodate this feature. In addition, Shiryaev solved the problem in the Bayes formulation [29], where the Bayes objective used was a means of stating the desire to minimize the expected delay to detecting a change while keeping the probability of false alarm or the expected number of false alarms before the change to some fixed value. Such an interpretation of the Bayes formulation suggests the usefulness of the BSDP for incorporating the numerous tradeoff issues of the decision process in FDI into a single design objective. In this way, the BSDP provides a conceptually simple design objective for the FDI problem, and it has been the basis of our research in the design of sequential decision rules. Although the optimal solution is impossible to calculate, the structure of the Bayes formulation provides a framework in which simple

suboptimal rules can be derived. We shall report our work along this direction in the remainder of this chapter and the next. In section 4.2, the Bayes formulation of the FDI problem is discussed. The nature of the optimal BSDR and the difficulties in its calculation are examined in Section 4.3 for ways of obtaining simple suboptimal rules. Chapter 5 contains the resulting approach to designing suboptimal sequential decision rules.

4.2 The Bayes Approach for FDI

In this section we will present a formulation of the FDI decision process as a BSDP, and we will discuss the advantages and limitations of the Bayes approach as a tool for designing decision rules for FDI.

In a sequential decision problem, the decision maker is allowed to make a sequence of observations. After each observation, he will decide whether to stop sampling and choose a terminal action, or to continue sampling and postpone the terminal decision to a later time. Hence the sequential decision rule consists of a stopping rule and a terminal decision rule which, in the FDI problem, are used to determine when to interrupt the residual-monitoring and what failure declaration to make, respectively. Each sequential decision rule leads to a particular performance tradeoff determined by the Bayes formulation. As we develop the Bayesian approach to FDI, we will see how the inherent tradeoff issues are incorporated into the formulation.

The BSDP formulation of the FDI problem consists of six elements:

1) Θ: the set of states of nature or failure hypotheses. An element θ of Θ may denote a single type i failure of size ν occurring at time τ ($\theta=(i,\tau,\nu)$) or the occurrence of a set of failures (possibly simultaneously), i.e. $\theta = \{(i_1, \tau_1,\nu_1),\ldots,(i_n,\tau_n,\nu_n)\}$. Although they do not add to the

structural complexity of the decision problem, multiple failures do increase the size of Θ and hence also increase the computational burden in the solution of the problem. In this study we will focus our attention on single failures for simplicity. Moreover, since failures occur infrequently, it is unlikely for failures to occur in rapid succession. Sufficient time is generally available for detecting and identifying a failure and re-starting the decision process before another failure occurs. From this vantage point, single failure indeed represent the dominant phenomenon.

In many applications it suffices to just identify the failure type without estimating the failure size. Then we may consider composite failure hypotheses of the form (i,τ) - a type i failure of any size occurring at time τ. Moreover, it is often true that a detection system based on $(i,\tau,\bar{\nu})$ for some appropriate $\bar{\nu}$ can also detect and identify the type of the failure (i,τ,ν) for $\nu \geq \bar{\nu}$. Thus, we may use $(i,\tau,\bar{\nu})$ to represent (i,τ). In the aircraft sensor FDI problem [11], for instance, excellent results were obtained using this approach. In situations where it is necessary to estimate ν in order to identify the failure type or choose a post-failure remedial action, a finite grid of failure magnitudes should provide sufficient resolution. In both cases, the failure size can be absorbed in the index i so that (i,τ) may represent a composite hypothesis or a failure of a certain magnitude. Now we have the discrete nature set

$$\Theta = \{(i,\tau),\ i=1,\ldots,M,\ \tau=1,2,\ldots,\} \qquad (4\text{-}1)$$

where we assume there are M different failure types of interest and any one of them may occur at time τ=1, or 2, or ... .

2) μ: the prior probability mass function (PMF) over the nature set Θ. This PMF represents the a priori information concerning the failure, i.e. how likely it is for each type of failure to occur, and when is a failure likely to occur. Because this information may not be available or accurrate in some cases, the need to specify μ is a drawback of the Bayes approach for such cases. Nevertheless, we will see that it can be regarded as a parameter in the design of a BSDR.

In general, μ may be arbitrary. Here, we will employ a special form of μ. We assume the underlying failure process has two properties: i) the M failures of Θ are independent of one another, and ii) the occurrence of each failure i is a Bernoulli process with (success) parameter ρ_i. The Bernoulli process (corresponding to the Poisson process in continuous time) is a common model for failures in physical components [30]; the independence assumption describes a large class of failures (such as sensor failures) while providing a simple approximation for the others. In this framework, the set Θ consisting of single failures only represents a subset, albeit a dominant one, of the exhaustive events defined over all possible outcomes including multiple failures. More precisely, Θ only describes the arrival of the first failure. Hence the PMF defined over Θ is a conditional PMF- $\mu(i,\tau)$ is the conditional probability that a failure will occur at time τ and that the failure will be of type i, given that this failure is the first ever to occur. Using the preceding reasoning, it is straightforward to show that

$$\mu(i,\tau) = \sigma(i)\rho(1-\rho)^{\tau-1} \qquad i=1,\ldots,M,\ \tau=1,2,\ldots, \tag{4-2}$$

where

$$\rho = 1 - \prod_{j=1}^{M} (1-\rho_j) \tag{4-3}$$

$$\sigma(i) = \rho_i(1-\rho_i)^{-1} \left[\sum_{j=1}^{M} \rho_j(1-\rho_j)^{-1} \right]^{-1} \tag{4-4}$$

The parameter ρ may be regarded as the parameter of the combined (Bernoulli) failure process - the occurrence of a (any) failure; $\sigma(i)$ can be interpreted as the marginal probability that the first failure is of type i. Note that (4-2) indicates the arrival of the first failure is memoryless, i.e.

$$\begin{aligned} \bar{\mu}(i,\tau \mid \text{no failure before } \tau_0) &= \sigma(i)\rho(1-\rho)^{\tau-\tau_0-1} \\ &= \mu(i,\tau-\tau_0), \quad i=1,\ldots,M \ \tau \geq \tau_0 \end{aligned} \tag{4-5}$$

This property will be useful in obtaining time-invariant suboptimal decision rules (see Chapter 5).

3) $\mathcal{D}(k)$: the discrete set of terminal actions (failure identifications) available to the decision maker when the residual-monitoring is stopped at time k. An element δ of $\mathcal{D}(k)$ may denote the pair (j,t), i.e. declaration of a type j failure to have occurred at time t. Alternatively, δ may represent an identification of the j-th failure type without regard for the failure time ($\delta = \bigcup_{t=1}^{k} (j,t)$), or it may signify the presence of a failure without specification of its type or time, i.e. simply an alarm ($\delta = \bigcup_{j=1,t=1}^{M,k} (j,t)$). Since the purpose of FDI is to detect and identify failures that have occurred, $\mathcal{D}(k)$ should only contain identifications that either specify failure times at or before k, or do not specify any failure time. As a result, the

number of terminal decisions specifying failures times grows with k while the number of decisions not specifying any time will remain the same. In addition, $\mathcal{D}(k)$ does not include the declaration of no failure, since the residual-monitoring is stopped only when a failure appears to have occurred. By continuing the residual-monitoring, it is understood that no (or insufficient) evidence has been gathered to declare any failure.

4) $L(k;\theta,\delta)$: the terminal decision cost (loss) function at time k defined over $\Theta \times \mathcal{D}(k)$. $L(k;\theta,\delta)$ denotes the penalty for deciding $\delta\in\mathcal{D}(k)$ at time k when the true state of nature is $\theta = (i,\tau)$. It is assumed to be bounded and non-negative and have the structure:

$$L(k;(i,\tau),\delta) = \begin{cases} L((i,\tau),\delta) & \tau\leq k, \quad \delta\in\mathcal{D}(k) \\ L_F & \tau>k \quad \delta\in\mathcal{D}(k) \end{cases} \tag{4-6}$$

where $L(\theta,\delta)$ is the underlying cost function that is independent of k; L_F denotes the penalty for a false alarm, and it may be generalized to be dependent on δ. It is only meaningful for a terminal action (identification) that indicates the correct failure (and/or time) to receive a lower decision cost than one that indicates the wrong failure (and/or time). We further assume that the penalty due to an incorrect identification of the failure time is only dependent on the error of such an identification. That is for $\delta = (j,t)$,

$$L((i,\tau),(j,t)) = L(i,j,(t-\tau)) \tag{4-7a}$$

and for δ with no time specification

$$L((i,\tau),\delta) = L(i,\delta) \tag{4.7b}$$

Clearly L provides the means for penalizing the various cross-detections according to how undesirable each of them is.

5) $r(k)$: the residual (observation) sequence. (We shall use $r(k)$ to denote both the random variable and its value, but the meaning will be clear from the context.) We shall assume $r(k) \in R^n$. The residual samples need not be independent and identically distributed in general, but their joint distribution is dependent on θ and is assumed to be known. We shall let $p(r(1),\ldots,r(k)\,|\,(i,\tau))$ denote their joint conditional density. Assuming that the residual is affected by the failure in a causal manner, its conditional density has the property

$$p(r(1),\ldots,r(k)\,|\,(i,\tau)) = p(r(1),\ldots,r(k)\,|\,(0,-)) \qquad i=1,\ldots,M, \quad \tau>k \tag{4-8}$$

where $(0,-)$ is used to denote the no-fail condition. In this research, we further assume that the residual is an independent Gaussian sequence with time-independent covariance function V(mxm matrix), i.e.

$$p(r(1),\ldots,r(k)\,|\,(i,\tau)) = \prod_{s=1}^{k} p(r(s)\,|\,(i,\tau)) \tag{4-9a}$$

$$p(r(k)\,|\,(i,\tau)) = \frac{1}{|V|^{1/2}(2\pi)^{m/2}} \exp\{-\frac{1}{2}[(r(k)-g(k;i,\tau))'V^{-1} \dot{x}(r(k)-g(k,i,\tau))]\} \tag{4-9b}$$

where $g(k;i,\tau)$ is the mean of the residual given that the failure (i,τ) has occurred. With the covariance assumed to be the same for all failures, the mean function $g(k;i,\tau)$, characterizes the effect of the failure (i,τ), and

it is henceforth called the signature of (i,τ), with $g(k;0,-)=0$ for the no-fail condition. For time-invariant systems $g(k;i,\tau)$ becomes

$$g(k;i,\tau) = \begin{cases} g_i(k-\tau) & k \geq \tau \\ 0 & k < \tau \end{cases} \qquad (4\text{-}10)$$

We have chosen to study residuals of the form (4-9) and (4-10) because their special structure facilitates the development of insights into the design of decision rules. Moreover, the Gaussian assumption is reasonable in many problems and has met with success in a wide variety of applications, e.g., [5] [11]. It should be noted that the use of more general probability densities for the residual, e.g. time-dependent V and signatures that depend on both k and τ ($g(i,k,\tau)$), will not invalidate the discussions in Section 4.3. The simple signature (4-9) and (4-10) considered here will facilitate the design of suboptimal rules (see Chapter 5).

6) $w(k,(i,\tau))$: the delay cost function having the properties:

$$w(k,(i,\tau)) = \begin{cases} w(i,k-\tau) \geq 0 & \tau \leq k \\ 0 & \tau > k \end{cases} \qquad (4\text{-}10a)$$

$$w(i,k_1-\tau) > w(i,k_2-\tau) \qquad k_1 > k_2 > \tau \qquad (4\text{-}10b)$$

After a failure has occurred at τ, there is a penalty for delaying the terminal decision until time $k>\tau$ with the penalty an increasing function of the delay $(k-\tau)$. In the absence of a failure, no penalty is imposed on

tne sampling. In this study we will consider a delay cost function that is linear in the delay, i.e.

$$c(i,k-\tau) = \begin{cases} c(i)(k-\tau) & \tau \leq k \\ 0 & \tau > k \end{cases} \qquad (4\text{-}11)$$

where c(i) is a positive function of the failure type i, and may be used to provide different delay penalties for different types of failures.

We have described the setting of the BSDP for the FDI decision process. The most important feature presented is the tradeoff structure provided by the terminal decision and delay cost functions. Generally, the more observations the decision maker has, the more certain he is about the true state of nature, and this will lead to a lower expected terminal decision cost which is due to false alarms and incorrect detections. On the other hand, he is penalized for the delay in decision that results from taking more observations. Hence, the cost functions L and w together form the basis for considering the tradeoffs among the various performance issues (detection delays, false alarms, etc.) simultaneously when the design objective is to minimize the total expected cost. Now we proceed to characterize sequential decision rules for minimizing the total cost, employing the approach of Ferguson [31].

A sequential decision rule naturally consists of two parts: a stopping rule (or sampling plan) and a terminal decision rule. The stopping rule, denoted by $\phi = (\phi(0), \phi(1;r(1)), \ldots, \phi(k;r(1),\ldots,r(k)),\ldots)$ is a sequence of functions of the observed residual samples, with $\phi(k;r(1),\ldots,r(k))=1$, or 0. When $\phi(k;r(1),\ldots,r(k))=1$, (0), residual-monitoring or sampling is stopped

(continued) after the k residual samples, $r(1),\ldots,r(k)$ are observed. Alternatively, the stopping rule may be defined by another sequence of functions $\Psi = (\psi(0),\psi(1;r(1)),\ldots,\psi(k;r(1),\ldots,r(k)),\ldots)$, where $\psi(k;r(1),\ldots,r(k))=1$ (0) indicates that residual-monitoring has been carried on up to and including time (k-1) and will (not) be stopped after time k when residual samples, $r(1),\ldots,r(k)$ are observed. The functions Φ and Ψ are related to each other in the following way

$$\psi(k;r(1),\ldots,r(k)) = \phi(k;r(1),\ldots,r(k)) \prod_{S=0}^{k-1} [1-\phi(s,r(1),\ldots,r(s))] \tag{4-12}$$

with $\psi(0) = \phi(0)$. The conditional probability of stopping at time k, given that the true state of nature is (i,τ), is

$$E_{i,\tau}\psi(k;r(1),\ldots,r(k)) = \int_{R^m x\ldots xR^m} \psi(k,r(1),\ldots,r(k))p(r(1),\ldots,r(k)\,|\,i,\tau)dr(1)\ldots dr(k) \tag{4-13}$$

and the probability, $P_s(i,\tau)$, of eventually stopping given $\theta = (i,\tau)$ is

$$P_s(i,\tau) = \sum_{k=0}^{\infty} E_{i,\tau}\psi(k;r(1),\ldots,r(k)) = 1 \tag{4-14}$$

If $P_s(i,\tau) \neq 1$ for all $(i,\tau) \in \Theta$, it is possible for the sampling to go on indefinitely even in the presence of a failure, and the expected delay cost will be infinite. Therefore, only stopping rules will $P_s(i,\tau) = 1$ are meaningful.

The terminal decision rule is a sequence of functions, $D = (d(0), d(1;r(1)), \ldots, d(k;r(1),\ldots,r(k)),\ldots)$, mapping residual samples, $r(1),\ldots,r(k)$ into the terminal action set $\mathcal{D}(k)$. The function $d(k;r(1),\ldots,r(k))$ represents the decision rule used to arrive at an action (identification) if sampling is stopped at time k and the residual samples, $r(1),\ldots,r(k)$ are observed. Actually, D only needs to be defined for those $r(1),\ldots,r(k)$ for which $\psi(k;r(1),\ldots,r(k)) = 1$. But it will become clear that it is useful to consider terminal decision rules independently of the stopping rule.

The FDI sequential decision procedure consists of two steps. According to the sampling plan, the decision maker determines if he is to continue the residual-monitoring. If he is to stop, he makes a failure identification according to the terminal decision rule. As a result of using the sequential decision rule (Φ,D), given (i,τ) is the true state of nature, the total expected cost is:

$$U_0[(i,\tau),(\Phi,D)] = \sum_{k=0}^{\infty} E_{i,\tau}\{\psi(k;r(1),\ldots,r(k))[c(k,(i,\tau)) + L(k;(i,\tau),d(k;r(1),\ldots,r(k)))]\} \qquad (4\text{-}15)$$

The BSDP is defined as: determine a sequential decision rule (Φ^*,D^*) so that the sequential Bayes risk U_s is minimized, where

$$U_s(\Phi,D) = EU_0[(i,\tau),(\Phi,D)] = \sum_{i=1}^{M}\sum_{\tau=1}^{\infty} \mu(i,\tau)U_0[(i,\tau),(\Phi,D)] \qquad (4\text{-}16)$$

(Φ^*,D^*) is called the Bayes Sequential Decision Rule (BSDR) with respect to μ, and it is optimal in the sense that it minimizes the sequential Bayes risk.

The BSDR can be determined by carrying out the minimization of (4-15) in two steps: first with respect to D, then with respect to Φ. While we postpone further discussions of the BSDR to the next section, we proceed to examine the Bayes risk closely for a better understanding of the BSDP.

Using (4-6) and (4-11), we can re-write U_0 as

$$U_0[(i,\tau),(\Phi,D)] = L_F \sum_{k=0}^{\tau-1} E_{i,\tau}\{\psi(k;r(1),\ldots,r(k))\} + c(i) \sum_{k=\tau}^{\infty} [(k-\tau)E_{i,\tau}\{\psi(k;r(1),\ldots,r(k))\}] + \sum_{k=\tau}^{\infty} [E_{i,\tau}\{\psi(k;r(1),\ldots,r(k))L(i,\tau),d(k;r(1),\ldots,r(k))\}] \tag{4-16}$$

In the following we will describe a special interpretation of the sequential risk for the FDI problem. Let us define the following notation

$$P_F(\tau) = \sum_{k=1}^{\tau-1} E_{i,\tau}\psi(k;r(1),\ldots,r(k)) \tag{4-17}$$

$$\mathcal{D} = \bigcup_{k=0}^{\infty} \mathcal{D}(k) \tag{4-18}$$

$$\tilde{S}(k,\delta)=\{[r(1),..,r(k)]:\psi(k;r(1),\ldots,r(k)=1,d(k,r(1),\ldots,r(k))=\delta\}, \quad \delta\in\mathcal{D} \tag{4-19}$$

$$P_r\{\tilde{S}(k,\delta)\,|\,i,\tau\} = \int_{\tilde{S}(k,\delta)} p(r(1),\ldots,r(k)\,|\,i,\tau)\,dr(1)\ldots dr(k) \tag{4-20}$$

where $P_F(\tau)$ is the probability of stopping to declare a failure before the failure occurs at τ, i.e. the probability of false alarm when a failure occurs at time τ; $\mathcal{D}$ is the set of terminal actions for all times; $\tilde{S}(k,\delta)$ is the region in the sample space of the first k residuals where the sequential rule (Φ,D) yields the terminal decision δ. Clearly, the $\tilde{S}(k,\delta)$'s are disjoint sets with respect to both k and δ, and we note that

$$E_{i,\tau}\psi(k;r(1),\ldots,r(k)) = \sum_{\delta\in\mathcal{D}(k)} P_r\{\tilde{S}(k,\delta)\,|\,i,\tau\} \tag{4-21}$$

Then (4-15) can be expressed as

$$U_0[(i,\tau),(\Phi,D)]=L_F P_F(\tau)+(1-P_F(\tau))\{c(i)\sum_{k=\tau}^{\infty}[(k-\tau)(1-P_F(\tau))^{-1}E_{i,\tau}\psi(k,r(1),..,r(k))] + \sum_{\delta\in\mathcal{D}} L((i,\tau),\delta)\sum_{k=\tau}^{\infty}[P_r\{\tilde{S}(k,\delta)\,|\,i,\tau\}(1-P_F)^{-1}\}$$

(4-22)

By (4-13), $(1-P_F(\tau))^{-1}E_{i,\tau}\psi(k;r(1),\ldots,r(k))$ for $k\geq\tau$ is the conditional probability that residual-sampling will be stopped at time k, given a type i failure has occurred at time τ and the sampling process has not been stopped before the failure occurred (i.e. no false alarm), and (4-21) then takes the form

$$U_0[(i,\tau),(\Phi,D)]=L_F P_F(\tau)+(1-P_F(\tau))[c(i)\bar{t}(i,\tau)+\sum_{\delta\in\mathcal{D}} L((i,\tau),\delta)P((i,\tau),\delta)] \tag{4-23}$$

where

$$\bar{t}(i,\tau) = \sum_{k=\tau}^{\infty} (k-\tau)(1-P_F(\tau))^{-1} E_{i,\tau}\psi(k;r(1),\ldots,r(k)) \qquad (4\text{-}24)$$

$$P((i,\tau),\delta) = \sum_{k=\tau}^{\infty} P_r\{\tilde{S}(k,\delta) \mid i,\tau\}(1-P_F)^{-1} \qquad (4\text{-}25)$$

The expression $\bar{t}(i,\tau)$ is the conditional expected delay in decision (i.e. stopping sampling and making a failure identification), given a type i failure has occurred at time τ and no false alarm has been signalled before this time. Similarly, $P((i,\tau),\delta)$ is the conditional probability of eventually declaring δ, given an i-th type failure has occurred at time τ and no false alarm has been signalled --$P((i,\tau),\delta)$ is the generalized cross-detection probability. Finally, the sequential Bayes risk U_s can be written as

$$U_s(\Phi,D) = \sum_{i=1}^{M} \sum_{\tau=1}^{\infty} \mu(i,\tau)\{L_F P_F(\tau)+(1-P_F(\tau))[c(i)\bar{t}(i,\tau)+ \sum_{\delta\in\mathcal{D}} L((i,\tau),\delta)P((i,\tau),\delta)] \qquad (4\text{-}26)$$

Equation (4-26) indicates that the sequential Bayes risk is a weighted combination of the conditional false alarm probability, expected delay to decision and cross-detection probabilities, and the optimal sequential rule (Φ^*,D^*) minimizes such a combination. From this vantage point, the cost functions (L and c) and the prior distribution (μ) provide for the weighting, hence, a basis for indirectly specifying the tradeoff relationships among the various performance issues. The advantage of the indirect approach is that only the total expected cost instead of every

individual performance issue needs to be considered explicitly in designing a sequential rule. The drawback of the approach, however, lies in the choice of a set of appropriate cost functions (and sometimes the prior distribution) when the physical problem does not have a natural set, as it doesn't in general. In this case, the Bayes approach is most useful with the cost functions (and the prior distribution) considered as design parameters that may be adjusted to obtain an acceptable design.

4.3 The Bayes Sequential Decision Rule

In this section, we will describe the optimal solution to the BSDP. Before we do that, it is instructive to examine the (unconditional) expected delay and terminal decision cost at time k, U(k), for a terminal action $\delta \in \mathcal{D}(k)$

$$U(k) = \sum_{i=1}^{M} \sum_{\tau=1}^{\infty} [c(k;(i,\tau)) + L(k;(i,\tau),\delta)]\mu(i,\tau)$$

$$= \sum_{i=1}^{M} \sum_{\tau=1}^{k} [c(i;(k-\tau) + L((i,\tau),\delta)]\mu(i,\tau) + L_F \sum_{j=1}^{M} \sum_{t=k+1}^{\infty} \mu(j,t)$$

$$= \sum_{i=1}^{M} \sum_{\tau=1}^{k} [c(i,k-\tau) + L((i,\tau),\delta)]\mu'(k;i,\tau) + L_F \mu'(k;0,-)$$

where the first equality follows from the definitions, the second one is a direct consequence of (4-6) and (4-10), and the last one follows from the notation

$$\mu'(k;i,\tau) = \begin{cases} \mu(i,\tau) & i=1,\ldots,M,\ \tau=1,\ldots,k \\ \sum_{j=1}^{M} \sum_{t=k+1}^{\infty} \mu(j,t) & i=0 \end{cases} \qquad (4\text{-}27)$$

where (0,-), i.e. i=0 is an artificial failure state representing the event of a failure occurring after the terminal decision time k. Alternatively (0,-) may be viewed as the no-fail state that has a dwindling prior probability as time progresses. Clearly, μ' is a time-dependent PMF, and it effectively defines a growing nature set, $\Theta(k) = \{(i,\tau),\ i=0,1,\ldots,M,\ \tau=1,..,k\}$, that corresponds to the increasing number of possible failure times. For failure-monitoring over the interval $[1,k_0]$, $\mu'(k_0;0,-)$ denotes the probability of no failure over the interval with (0,-) as the no-fail state. Since they can be used interchangeably in the calculation of the expected cost, (Θ,μ) and $(\Theta(k),\mu'(k))$ are equivalent for the BSDP. But the latter will be used, due to the resulting simplification in notation. We now proceed to describe the BSDR.

It is clear that the expected delay cost is independent of what terminal decision is made. Thus, once sampling is stopped the optimal terminal decision rule must be that which minimizes the expected decision cost. This implies that the Bayes terminal decision rule D^* is independent of the stopping rule and is a sequence of fixed-sample-size (FSS) Bayes rules [32]. Therefore, $D^* = (d^*(0), d^*(1;r(1)),\ldots,d^*(k;r(1),\ldots,r(k)),\ldots)$ where $d^*(k)$ is the k-sample Bayes rule (with respect to $\mu'(k)$) that minimizes the FSS Bayes risk:

$$\pi(d(k)) = \int_{R^m \times\ldots\times R^m} \sum_{i=0}^{M} \sum_{\tau=1}^{k} \mu'(k;i,\tau) L((i,\tau),d(k)) p(r(1),..,r(k)\,|\,i,\tau)\, dr(1)\ldots dr(k) \qquad (4\text{-}28)$$

Hence, the k-sample Bayes rule $d^*(k)$ also minimizes the integrand of (4-28), By simple manipulations, the Bayes rule can be expressed in terms of the

likelihood ratios Λ or the posterior probabilities q of the nature states given the residual samples, r(1),...,r(k) [32]:

$$d^*(k) = \arg\min_{d(k)} \sum_{i=0}^{M} \sum_{\tau=1}^{k} L((i,\tau),d(k))\mu'(k;i,\tau)\Lambda(k;i,\tau) \tag{4-29}$$

or

$$d^*(k) = \arg\min_{d(k)} \sum_{i=0}^{M} \sum_{\tau=1}^{k} L((i,\tau),d(k))q(i,\tau|k) \tag{4-30}$$

where, assuming $p(r(1),...,r(k)|0,-) \neq 0$,

$$\Lambda(k;i,\tau) = \frac{p(r(1),...,r(k)|i,\tau)}{p(r(1),...,r(k)|0,-)} \tag{4-31}$$

and

$$q(i,\tau|k) = \frac{\mu'(k;i,\tau)p(r(1),...,r(k)|i,\tau)}{\sum_{j=0}^{M} \sum_{t=1}^{k} \mu'(k;j,t)p(r(1),...,r(k)|j,t)} \tag{4-32}$$

The Bayes rule given in (4-29) is in the form of a likelihood ratio test. In some cases, it is more convenient to work with the log likelihood ratios $\tilde{L}(k;i,\tau)$:

$$\tilde{L}(k;i,\tau) = \ell n\, \Lambda(k;i,\tau) \tag{4-33}$$

and (4-29) can be transformed accordingly. In general, the FSS decision rule divides the residual sample space into terminal decision regions $\{T(k,\delta),\ k=1,2,...\}$ such that $d(k;r(1),...,r(k))=\delta \in \mathcal{D}(k)$ if $(r(1),...,r(k)) \in T(k,\delta)$. Then (4-28) can be re-written as

$$\pi(d(k)) = \sum_{i=0}^{M} \sum_{\tau=1}^{k} \mu'(k;i,\tau) \sum_{\delta \in \mathcal{D}(k)} L((i,\tau),\delta) Q(k;i,\tau,\delta) \tag{4-34}$$

where

$$Q(k;i,\tau,\delta) = \int_{T(k,\delta)} p(r(1),\ldots,r(k))\,dr(1),\ldots,dr(k) \tag{4-35}$$

Note that

$$\sum_{\delta \in \mathcal{D}(k)} Q(k;i,\tau,\delta) = 1 \tag{4-36}$$

The quantity $Q(k;i,\tau,\delta)$ has the interpretation of a k-sample detection probability (i.e. probability of deciding δ based on k observations when (i,τ) is true) of which the probabilities of cross detection, correct detection, etc. of the k-sample decision problem are special cases. From (4-34) we see that the Bayes rule $d^*(k)$ minimizes a linear combination of the detection probabilities where the decision cost function and the prior probability μ' play the role of weights.

Now we will turn our attention to the optimal stopping rule. First we will consider stopping rules that terminate sampling at or before time $k>0$, i.e.

$$\sum_{k=1}^{K} \psi(k;\, r(1),\ldots,r(k)) = 1 \tag{4-37}$$

A sequential decision problem using such a rule is said to be truncated at time K. Then, the optimal stopping rule for the non-truncated problem can be obtained from the optimal truncated rule by letting $K \to \infty$. The optimal truncated rule can be determined via a straightforward application of the

principle of optimality [33]*, and we will state the result (which is contained in [22] and [31], for example) in the following.

Let us define J(k) to be the total expected cost of stopping at time k and applying the optimal terminal decision rule $d^*(k)$, given that $r(1),\ldots,r(k)$ have been observed:

$$J(k) = \sum_{i=1}^{M} \sum_{\tau=1}^{k} q(k;i,\tau|k)\,[c(i)(k-\tau)+L(i,\tau),\delta^*)] + q(k;0,-|k)L_F \tag{4-38}$$

where

$$\delta^* = d^*(k;r(1),\ldots,r(k)) \tag{4-39}$$

$$q(k;i,\tau|k) = \frac{\mu'(k;i,\tau)p(r(1),\ldots,r(k)|i,\tau)}{\sum_{j=1}^{M}\sum_{t=1}^{k}\mu'(k;j,t)p(r(1),\ldots,r(k)|j,t)} \tag{4-40}$$

The minimum expected cost-to-go at time k, $\tilde{J}_K(k)$, for the decision problem truncated at K is given by

$$\tilde{J}_K(k-1) = \min[J(k-1),E\{\tilde{J}(k)|r(1),\ldots,r(k)\}],\ k=1,\ldots,K-1 \tag{4-41a}$$

$$\tilde{J}_K(K) = J(K) \tag{4-41b}$$

Note that both J(k) and $\tilde{J}_K(k)$ are functions of $r(1),\ldots,r(k)$.

---

* The principal of optimality: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

The optimal stopping rule for the truncated problem is

$$\phi_K^* = (\phi_K^*(1;r(1)),\ldots,\phi_K^*(k;r(1),\ldots,r(k)),\ldots)$$

where

$$\phi_k^*(k;r(1),\ldots,r(k)) = \begin{cases} 1 & \text{if } J(k) < E\{\tilde{J}(k+1)\,|\,r(1),\ldots,r(k)\} \\ 0 & \text{otherwise} \end{cases} \qquad (4\text{-}42a)$$

$$\phi_k^*(K;r(1),\ldots,r(K)) = 1 \qquad (4\text{-}42b)$$

That is, sampling is terminated at time k if the total conditional expected cost (J(k)) of stopping at k to use the optimum terminal decision rule, given that r(1),...,r(k) have been observed, is lower than that ($E\{\tilde{J}_k(k+1)\,|\,r(1),\ldots,r(k)\}$) of taking an additional sample and using the optimum sequential rule from them on. The term $\tilde{J}_K(k)$ is the total expected cost of using the optimal stopping rule (for the problem truncated at K) and the optimum terminal decision rule at times k, k+1,..., and k, given that r(1),...,r(k) have been observed. Hence, it is called the optimal expected-cost-to-go at k. The sequential Bayes risk $U_s(\phi_K^*,D^*)$ for the truncated problem is simply

$$U_s(\phi_K^*,D^*) = \tilde{J}_K(0), \quad k=0,1,.., \qquad (4\text{-}43)$$

and we will let $\tilde{J}_\infty(0)$ denote the (finite) Bayes risk $U_s(\phi^*,D^*)$ that is associated with the optimal non-truncated sequential rule (ϕ^*,D^*), i.e. the BSDR.

The set of all stopping rules that terminate sampling at or before time k contains the set of all stopping rules that terminate sampling at or before time k-1. Therefore, we have the sequence of inequalities

$$\tilde{J}_o(0) \geq \ldots \geq \tilde{J}_k(0) \geq \tilde{J}_{k+1}(0) \geq \ldots \tag{4-44}$$

Due to the fact that the terminal decision cost function L is bounded, it can be shown that $\tilde{J}_K(0) \to \tilde{J}_\infty(0)$ as $K \to \infty$ (see Appendix A). Consequently, the optimal sequential rule (ϕ^*, D^*) for the non-truncated problem is the limit of (ϕ^*, D^*) as $K \to \infty$, and the former can be approximated by the latter for a sufficiently large K.

Note that the determination of the optimal truncated stopping rule requires solving (4-41) backwards in time. In fact, (4-41) and (4-42) describe a dynamic programming problem for which the solution is extremely difficult to calculate due to the immense storage and computation required [17]. Consequently, the optimal stopping rule is generally impossible to compute, and suboptimal rules must be used.

Despite the impractical nature of its solution, the BSDP provides a useful framework for designing suboptimal decision rules for the FDI problem because of its inherent characteristic of explicitly weighing the tradeoffs between detection speed and accuracy (in terms of the cost structure). In the previous section we saw that a sequential decision rule defines a set of sequential decision regions $\tilde{S}(k,\delta)$,* and the decision regions corresponding to

* Since the posterior probabilities q, the likelihood ratios Λ, and the log likelihood ratios $\tilde{l}$ are alternative sets of sufficient statistics, it is easy to see that a sequential rule also defines sequential decision regions in the space of each of these set of decision statistics.

the BSDR yield the minimum risk. From this vantage point, the design of a suboptimal rule can be viewed as the problem of choosing a set of decision regions that would yield a reasonably small risk. This is the essence of the approach to suboptimal rule design that we will describe in Chapter 5.

CHAPTER 5

SUBOPTIMAL SEQUENTIAL DECISION RULES

From the previous chapter we see that the Bayes formulation of the FDI decision problem is a suitable one because of its built-in performance tradeoff structure. Although the optimal rule (the BSDR) is computationally intractable, practical, suboptimal rules with good performance may be determined using the Bayes framework. This chapter is devoted to the discussion of our approach to the design of such suboptimal rules for FDI. While it covers a wide range of issues, this discussion, by far, does not exhaust all possibilities. Rather, it will serve to demonstrate how this framework can be useful for the systematic approach to decision rule designs.

In Section 5.1 we will first examine an approximation scheme for the BSDR that is directed at alleviating its overwhelming computational requirements. The resulting simplified decision rule will provide the basic form for a range of suboptimal rules of varying degrees of complexity. As we describe these rules we will also examine their computational structure so as to assess their practicality. In Section 5.2 we will consider the risk evaluation for some simple suboptimal rules. An algorithm for approximating the risk will also be described. The choice of design parameters and the risk-minimization procedure will be discussed in Section 5.3, and a summary of the design methodology described in Section 5.1 - 5.3 is included in Section 5.4. Our experience with this decision rule design methodology through the study of numerical examples and simulations is reported in the next chapter.

5.1 Suboptimal Rules Based on the BSDR

5.1.1 The Sliding Window Approximation

The immense computation associated with the BSDR is partly due to the increasing number of failure hypotheses as time progresses. In fact, this phenomenon (often called the "growing-bank-of-filters") is common to detection schemes, such as the GLR [5], where the failure time is explicitly taken into account in the failure hypotheses. The remedy for the problem studied here, as in [5] for instance, is the use of a sliding window to limit the number of failure hypotheses to be considered at each time. The application of the sliding window approximation to the BSDR for the infinite time horizon problem yields a sequential decision rule that uses only a sliding window of a fixed number of residual samples. This brings a saving in the storage of residual samples, as the BSDR, in contrast requires all past samples. Furthermore, such a sequential rule is independent of time after a window-full of residual samples has been gathered, while the BSDR is a time-dependent rule. Because of these desirable simplifications, the sliding window scheme has become the backbone of our study of the design of suboptimal rules. We now proceed to describe this approximation scheme.

The only assumption made under the sliding window approximation is that essentially all failures can be detected within W time steps after they have occurred, or that if a failure is not detected within this time it will not be detected in the future. Thus, when we have progressed to time k we

can be quite confident that no failure has occurred before time k-W+1, and we only need to consider the possibility of a failure occurring at some $\tau \geq k-W+1$. Consequently we have modified the nature set and terminal decision set at time k to be $\Theta^W(k)$ and $\mathcal{D}^W(k)$, respectively: $\Theta^W(k) = \{(i,\tau),\ i=1,\ldots,M,\ \tau \geq k-W+1\}$, and $\mathcal{D}^W(k)$ contains all the elements of $\mathcal{D}(k)$ that either specify failure times in the interval $[k-W+1,k]$, or do not specify any time at all. The prior probability mass function $\bar{\mu}^W(k;i,\tau)$ defined over $\Theta^W(k)$ may be regarded as the conditional probability that a type i failure will occur at time $\tau \geq k-W+1$, given that no failure has occurred before k-W+1. Using the memoryless nature of $\bar{\mu}$ (see Section 4.2), $\bar{\mu}^W$ can be easily shown to be

$$\bar{\mu}^W(k;i,\tau) = \begin{cases} 0 & \tau < k-W+1 \\ \sigma(i)\,\rho(1-\rho)^{\tau-k+W-1} & \tau \geq k-W+1 \end{cases}$$

For $k_1,\ k_2 \geq W$, the triplet $\{\Theta^W(k),\ \bar{\mu}^W(k),\ \mathcal{D}^W(k)\}$ for $k=k_1$ is clearly a time-shifted version of that for $k=k_2$. It is convenient to define a new time variable $\bar{\tau}$ that is related to the failure time τ by $\bar{\tau} = k-\tau$; $\bar{\tau}$ has the interpretation as the failure time relative to the decision time k, i.e. a positive (negative) $\bar{\tau}$ indicates a failure time that is $|\bar{\tau}|$ step before (after) k. Using this notation, at each time k, we have the nature set, prior probability mass function, and the terminal decision set in the following forms:

$$\Theta^W = \{(i,\bar{\tau}),\ i=1,\ldots,M,\ \bar{\tau}=W-1,W-2,\ldots,\} \tag{5-1}$$

$$\bar{\mu}^W(i,\bar{\tau}) = \begin{cases} 0 & \bar{\tau}\geq W \\ \alpha(i)\rho(1-\rho)^{W-\bar{\tau}-1} & \bar{\tau}<W \end{cases} \tag{5-2}$$

$$\mathcal{D}^W = \{\delta \text{with time specifications relative to k that are not earlier than k-W+1}\} \tag{5-3}$$

Let us recall three useful properties of the BSDP under investigation: i) the only time dependence of the failure signatures is manifested through the dependence on the elapsed times since the onset of failures, ii) the cost of incorrect failure time identification is a function of the difference between the true and declared times, and iii) the decision delay cost is proportional to the delay. It is now clear that the terminal decision problem at different times beyond W are time-shifted versions of the problem defined by $\{\Theta^W, \bar{\mu}^W, \mathcal{D}^W, L, g\}$. Consequently the terminal decision rule d^W mapping the sliding window of residuals $[r(k-W+1),..,r(k)]$ into $\mathcal{D}^W$ is a W-sample (Bayes) rule that is the same for all $k\geq W$. Similarly, the stopping problems encountered at different stages of the sequential decision process are defined by the same six elements: $\{\Theta^W, \mu^W, \mathcal{D}^W, L, W, g\}$, and the stopping rule ϕ^W defined over the sample space of the sliding window of residuals $[r(k-W+1),\ldots,r(k)]$ is the same for all $k\geq W$. Therefore, the sequential rule (ϕ^W, d^W) is much simpler than the BSDR which consists of a

infinite sequence of time dependent rules. In addition, only a finite storage is required by (ϕ^W, d^W) for the window of data as opposed to the growing storage needed by the BSDR. Because of these desirable features, the sliding window approximation will be the basis of our suboptimal rule designs.

Since the sequential rule (ϕ^W, d^W) uses a sliding window of residuals (hence called a "sliding window sequential decision rule"), it requires mandatory sampling through the initial W steps in order to fill up the data window. One minor drawback of such a feature is increased delays in detecting failures occurring within the first W time steps. Fortunately, all practical window sizes are reasonably small so that the probability of a failure occurring within the first W time steps is negligible, and the above mentioned detection delays will not have any significant impact on the overall performance of the sliding window rule.

A more important design aspect introduced by the use of a window rule is the tradeoff between detection performance and computational complexity. A window rule with a long window is more likely to deliver good detection accuracy than one with a short window, because with a long window, more data is used and more possible failure times can be considered. But on the other hand, a long window rule requires more computations for both the off-line performance evaluation during the design process and the on-line processing of the window of data to generate the decisions. From our vantage point, the window size W is considered along with the prior probability μ and the cost functions L and w as design parameters within the Bayes formulation that may

be adjusted to achieve a satisfactory sliding window decision rule. This will be discussed further in subsection (5.3.2.) The Bayes design problem now becomes: for a set of μ, W, L and c, find a (ϕ^W, d^W) that minimizes the sequential risk $U_s(\phi^W, d^W)$. As it stands, the solution of this problem still requires a tremendous amount of computation, albeit much less than that required for the BSDR. In the next subsection we will examine the computational structure associated with the risk evaluation for sliding window rules in order to indicate now how further simplifications may be introduced.

5.1.2 Sliding Window Sequential Decision Rules

Similar to the BSDR, the window rule (ϕ^W, d^W) divides the sample space of the sliding window of residuals, or equivalently, the space of vectors of posterior probabilities (q), likelihood ratios (Λ), or log likelihood ratios ($\tilde{L}$) of the sliding window of failure hypotheses into disjoint sequential decision regions $\{S_0, S_1, \ldots, S_N\}$. Because the residuals are assumed to be Gaussian variables, the log likelihood ratios are simpler to work with than the likelihood ratios or the posterior probabilities. We will only use the log likelihood ratios as the decision statistics. Now suppose there are N elements in $\mathcal{D}^W$ and these elements are indexed such that $\mathcal{D}^W = \{S_1, \ldots, S_N\}$. In terms of the sequential decision regions defined in the $\tilde{L}$-space, the sliding window rule states: At each time $k \geq W$, we form the decision statistics $\tilde{L}(k)$ from the window of residual samples. If $\tilde{L}(k) \in S_i$, for i=1,.., or N, we will stop sampling to declare δ_i; otherwise, $\tilde{L}(k) \in S_0$, and we will proceed to take one more observation of the residual. The

Bayes design problem is to determine a set of regions $\{S_0^*, S_1^*, \ldots, S_N^*\}$ that minimizes the sequential risk $U_s^W(\{S_i\})$, i.e.

$$\{S_i^*\} = \arg \min_{\{S_i\}} U_s^W(\{S_i\}) \tag{5-4}$$

Expression (5-4) represents a functional minimization problem for which a solution is generally very difficult to determine. A simpler alternative to this problem is to constrain the decision regions to take on special shapes, $\{S_i(f)\}$, that are parameterized by a fixed dimensional vector, f, of design variables. Then the resulting design problem involves the determination of a set of parameter values f^* that minimizes the risk $U_s^W(\{S_i(f)\})$

$$f^* = \arg \min_{f} U_s^W(\{S_i(f)\}) \tag{5-5}$$

In this study, we will focus our attention on a special set of parametrized sequential decision regions, because they are simple and they serve well to illustrate that the Bayes formulation can be exploited, in a systematic fashion, to obtain simple suboptimal rules that are capable of delivering good performance. Next, we shall describe this set of simple decision regions.

The window of failure hypotheses consists of $\Theta^W = \{(i,\bar{\tau}),\ i=0,1,\ldots,M,\ \bar{\tau}=W-1,W-2,\ldots\}$, where $(0,-)$ denotes the hypothesis of no failure in the window. Suppose the terminal decision set is of the form $\mathcal{D}^W = \{(j,\bar{t}),\ j=1,\ldots,M,\ \bar{t}=0,\ldots,W-1\}$ with $(j,\bar{t})$ corresponding to

declaring a type j failure occurring at time $k-\bar{t}$. The sequential decision regions we will study are of the form:

$$S(j,\bar{t}) = \{\tilde{L}(k): \tilde{L}(k;j,\bar{t}) > f(j,t)$$
$$e^{-1}(j,\bar{t})\,[\tilde{L}(k;j,\bar{t}) - f(j,\bar{t})] > e^{-1}(i,\bar{\tau})\,[\tilde{L}(k;i,\bar{\tau}) - f(i,\bar{\tau})],$$
$$(i,\bar{\tau}) \neq (j,\bar{t})\} \qquad (5\text{-}6a)$$

$$S(0,-) = \{\tilde{L}(k): \tilde{L}(i,\bar{\tau}) \leq f(i,\bar{\tau}),\ i=1,\ldots,M,\ \bar{\tau}=0,\ldots,W-1\} \qquad (5\text{-}6b)$$

where $\tilde{L}(k) = [\tilde{L}(k;1,0),\ldots,\tilde{L}(k;M,W-1)]'$; $S(j,\bar{t})$ is the stop-to-declare-$(j,\bar{t})$ region, and $S(0,-)$ is the continue region (see Figure 5-1 for an illustration in two dimensions). Note that this set of decision regions may be easily modified to accomodate the case where $\mathcal{D}^W$ has some of its elements replaced by a composite decision, e.g., if $\{(j,0),..,(j,W+1)\}$ is replaced by $\delta = \bigcup_{\bar{t}=0}^{W-1} (j,\bar{t})$ (i.e. declaring a type j failure without regard of the failure time), we have the stop-to-declare-δ region $S(\delta) = \bigcup_{\bar{t}=0}^{W-1} S(j,\bar{t})$. In (5-6) the f's are known as the decision thresholds, and the ϵ's are the normalization constants. (As shown in Figure 5-1 for the 2-dimensional case, the ϵ's determine the slope of the boundary between two stopping regions). Generally, the ϵ's may be regarded as design parameters along with the f's. In this study $\epsilon(i,\bar{\tau})$ is simply taken to be the standard deviation of $\tilde{L}(k;i,\bar{\tau})$.

Recall that the residual samples are Gaussian variables, Then the log likelihood ratio $\tilde{L}(k;i,\bar{\tau})$ is given by:

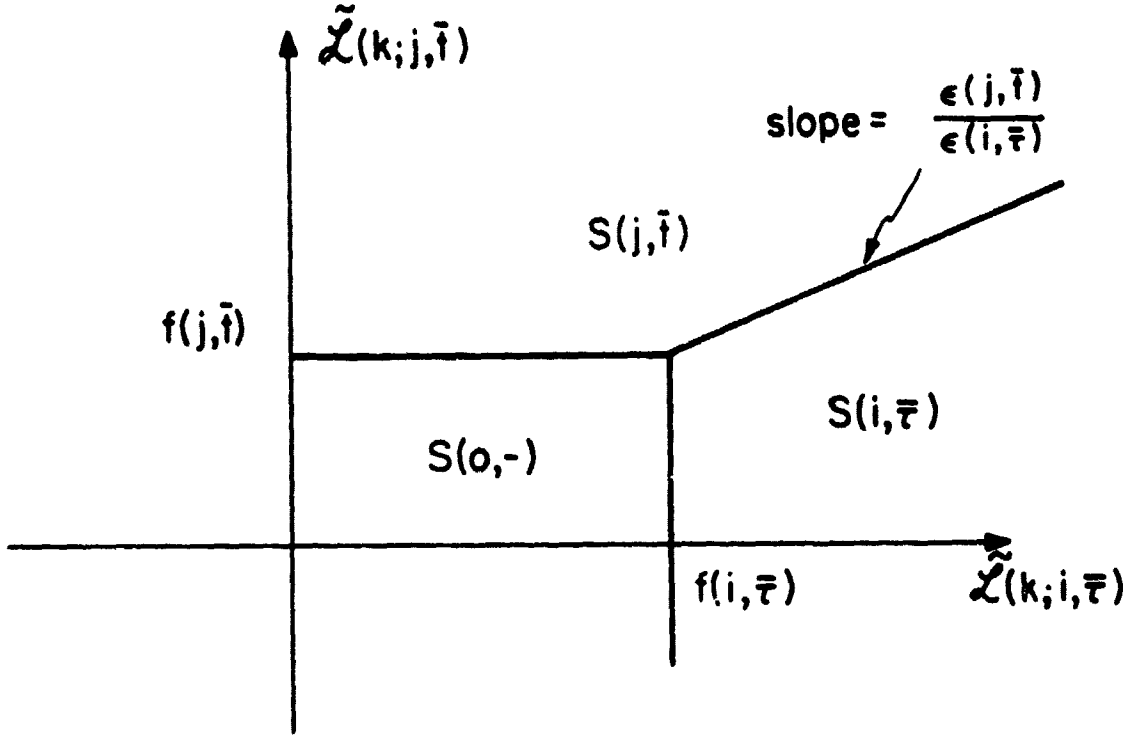


FIGURE 5-1: Sequential Decision Regions in Two Dimensions.

$$\tilde{L}(k;i,\bar{\tau}) = \sum_{s=0}^{\bar{\tau}} g_i'(s)V^{-1}r(k-\bar{\tau}+s) - \frac{1}{2}\sum_{s=0}^{\bar{\tau}} g_i'(s)V^{-1}g_i(s) \qquad (5\text{-}7)$$

Since the second term in (5-7) is a function of i and $\bar{\tau}$ and is constant for all $L(k;i,\bar{\tau})$, k=W, W+1,..., it may be absorbed into $f(i,\bar{\tau})$ in the definition of the decision regions. As a result, the decision region may be re-defined in terms of a set of new decision statistics, $L(k)$:

$$S(j,\bar{t}) = \{L(k): L(j,t)>f(j,t)$$
$$e^{-1}(j,\bar{t})[L(k;j,\bar{t})-f(j,\bar{t})]>e^{-1}(i,\bar{\tau})[L(k;i,\bar{\tau})-f(i,\bar{\tau}),$$
$$(i,\bar{\tau})\neq(j,\bar{t})\} \qquad (5\text{-}8a)$$

$$S(0,-) = \{L(k): L(k;i,\bar{\tau})\leq f(i,\bar{\tau}),\ i=1,\ldots,M,\ \bar{\tau}=0,\ldots,W-1\} \qquad (5\text{-}8b)$$

where

$$L(k;i,\tau) = \sum_{s=0}^{\bar{\tau}} g_s'(s)V^{-1}r(k-\tau+s) \qquad (5\text{-}9)$$

and $f(i,\bar{\tau})$ has absorbed the constant term in (5-7). The decision statistics $L(k)$ can be viewed as the state of a linear time invariant system driven by the residual. To see this, we will define the following notations:

$$L_{\tau}(k) = [L(k;1,\bar{\tau}),\ldots,L(k;M,\bar{\tau})]',\ \bar{\tau}=0,\ldots,W-1 \qquad (5\text{-}10a)$$

$$L(k) = [L_0(k),\ldots,L_{W-1}(k)]' \qquad (5\text{-}10b)$$

$$G_{\bar{\tau}} = \begin{bmatrix} g_1'(\bar{\tau})V^{-1} \\ \vdots \\ g_M'(\bar{\tau})V^{-1} \end{bmatrix} \qquad \bar{\tau}=0,\ldots,W-1 \tag{5-11a}$$

$$G = \begin{bmatrix} G_0 \\ \vdots \\ G_{W-1} \end{bmatrix} \tag{5-11b}$$

$$J = \begin{bmatrix} 0 & \cdots & 0 \\ I & \ddots & \vdots \\ & \ddots & \\ 0 & I & 0 \end{bmatrix} \tag{5-12}$$

Then, from the definition of $L(k)$, we have

$$L(k+1) = JL(k) + Gr(k+1) \qquad k=0,1,\ldots \tag{5-13a}$$

$$L(0) = 0 \tag{5-13b}$$

Note that $L(k)$ is of dimension MW, and it is also a Markov process under any failure hypothesis.

With the sequential decision regions defined, we are now ready to examine their associated risk. First, it is convenient to define $S_0(k)$

to be the event that no failure declaration has been made up to and including time k, i.e.

$$S_0(k) = \{L(k) \in S(0,-),\ L(k-1) \in S(0,-), \ldots, L(W) \in S(0,-)\} \tag{5-14}$$

Since sampling through the first W time steps is mandatory, it is not necessary for (5-13) to include $S(k) \in S(0,-)$ for $k<W$. Using the sequential rule defined by (5-8) in the risk expression (4-26), we get

$$U_s^W(f) = L_F \sum_{i=1}^{M} \sum_{\tau=W+1}^{\infty} \mu(i,\tau) \sum_{k=W}^{\tau-1} \sum_{j=1}^{M} \sum_{\bar{t}=0}^{W-1} P_r\{L(k) \in S(j,\bar{t}),\ S_0(k-1) \mid 0,-\}$$

$$+ \sum_{i=1}^{M} \sum_{\tau=1}^{\infty} \mu(t,\tau) \sum_{k=\max[W,\tau]}^{\infty} \sum_{j=1}^{M} \sum_{\bar{t}=0}^{W-1} [c(i)(k-\tau) + L(i,j,(k-\bar{t}-\tau)]$$

$$\times \Pr\{L(k) \in S(j,\bar{t}),\ S_0(k-1) \mid i,\tau\} \tag{5-15}$$

where we have used $U_s^W(f)$ to denote the sequential risk due to a set of sequential decision regions with window size W and parameterized by f. Note that the mandatory continuation of the sampling process through the first W steps is reflected in the lower summation limits for τ and k.

To evaluate $U_s^W(f)$, we need to determine the set of probabilities, $\{\Pr\{L(k) \in S(j,\bar{t}),\ S_0(k) \mid i,\tau\}, k \geq W,\ j=0,1,\ldots,M,\ \bar{t}=0,\ldots,W-1\}$, which, indeed, is the goal of many research efforts in the so-called level-crossing problem [34]. Unfortunately, useful results (bounds and approximations of such probabilities) are only available for the scalar case [35],[36],[37], i.e. in terms of our problem, $L(k)$ is a scalar and the decision regions become

intervals on the real line, and thus are not applicable to our problem. For the general multidimensional problem, we presently have to resort to numerical methods. As it stands, each of the probabilities is an integral of a kMW-dimensional Gaussian density over the compound region $S(0,-)x...xS(0,-)xS(j,\bar{t})$, which, for large kMW, becomes extremely unwieldy and difficult to evaluate. However, the common structure of the probability events --the $S_0(k-1)$ of the event $\{L(k) \in S(j,t),\ S_0(k-1)\}$, may be exploited to obtain a more tractable computation structure for the probabilities. To accomplish this, we can use Bayes rule to arrive at an recursive expression for the probabilities:

$$P(L(k+1)\,|S_0(k),i,\tau) = \left[\int_{S(0,-)} P(L(k)\,|S_0(k-1),i,\tau)\,dL(k)\right]^{-1} \times \int_{S(0,-)} P(L(k+1)\,|L(k),S_0(k-1),i,\tau)\,P(L(k)\,|S_0(k-1),i,\tau)\,dL(k) \qquad k>W \tag{5-16}$$

$$Pr\{L(k) \in S(j,\bar{t}),S_0(k-1)\,|i,\tau\} = Pr\{S_0(k-1)\,|i,\tau\} \int_{S(j,\bar{t})} p(L(k)\,|S_0(k-1),i,\tau)\,dL(k), \qquad j=0,1,...,M,\ \bar{t}=0,...,W-1 \tag{5-17}$$

with

$$Pr\{L(W) \in S(j,t)\,|i,\tau\} = \int_{S(j,\bar{t})} p(L(W)\,|i,\tau)\,dL(W), \qquad j=0,1,...,M,\ \bar{t}=0,...,W-1 \tag{5-18}$$

where $p(L(k+1)\,|S_0(k),i,\tau)$ denotes the conditional probability density of

$L(k+1)$, given $S_0(k)$ and that a type i failure has occurred at time τ; $p(L(k+1)|L(k), S_0(k-1), i, \tau)$ is the density for the transition from $L(k)$ to $L(k+1)$, given $S_0(k-1)$ and (i,τ); $p(L(W)|i,\tau)$ is the (Gaussian) density for $L(W)$ under a hypothesis (i,τ). Using (5-16)-(5-18), we have to contend with MW-dimensional integrals instead of integrals with increasing dimensions as required by the straightforward evaluation of the probabilities. Nevertheless, this problem is still difficult, since even for M=2 and W=10 the integrals are 20-dimensional.

In the remainder of this section we will examine the computational complexity associated with the evaluation of (5-16)-(5-18). First, we will consider the transition density in (5-16). Noting that $L(k)$ is a Markov process we have

$$p(L(k+1)|L(k), S_0(k-1), i, \tau) = p(L(k+1)|L(k), i, \tau) \tag{5-19}$$

From (5-19), we have

$$L(k+1) = JL(k) = Gr(k+1) \tag{5-20}$$

The dimension of $L(k)$, MW, is generally greater the rank of G, which is assumed to be $m \geq M$ (m is the dimension of r). The increment $L(k+1) - JL(k)$ is due to r(k+1) and can only lie in, at best, m-dimensional subspace of R^{MW}. That is, the one-step transition density in (5-19) is degenerate. Then, using the fact that r(k+1) and $L(k)$ are independent of each other, it can be easily shown that

$$\begin{aligned} &p(L(k+1)|L(k), i, \tau) \\ &\quad = u_0(||[I-G(G'G)^{-1}G][L(k+1)-JL(k)]||)\,|G'G|^{-1} \\ &\qquad\qquad \times\ p(r(k+1)=(G'G)^{-1}G'[L(k+1)-JL(k)]|i,\tau) \end{aligned} \tag{5-21}$$

where u_0 is the impulse function, $|G'G|$ is the determinant of G'G, and $p(r(k+1)|i,\tau)$ denotes the Gaussian density of $r(k+1)$ under (i,τ). As expected, the transition density (5-21) for a given $L(k)$ is zero for some $L(k+1)$, namely those values such that $(L(k+1)-JL(k))$ cannot be accounted for with any $r(k+1)$, i.e. when $[I-G(G'G)^{-1}G'][L(k+1)-JL(k)]\neq 0$.

Finally, the recursive equation (5-16) for the conditional density of L can be re-written as:

$$p(L(k+1)\ S_0(k),i,\tau) = \left[\int_{S(0,-)} p(L(k)|S_0(k-1),i,\tau)\,dL(k)\right]^{-1}$$

$$\times \int_{S(0,-)\cap F(L(k+1))} |G'G|^{-1} p(r(k+1)=(G'G)^{-1}G'[L(k+1)-JL(k)]|i,\tau)\, p(L(k)|S_0(k-1),i,\tau)\,dL(k) \qquad k>W \qquad (5\text{-}22)$$

where

$$F(L(k+1)) = \{L(k):[I-G(G'G)^{-1}G'][L(k+1)-JL(k)] = 0\} \qquad (5\text{-}23)$$

The set $F(L(k+1))$ is the set of all $L(k)$'s that together with some $r(k+1)$ will produce the given $L(k+1)$. If rank $G_0 < M$ (see (5-11)), the pair (J,G) representing the system (5-13) is uncontrollable, and there are some values of L that cannot be generated by (5-13). This corresponds to the case, for example, when two failure hypotheses represent different magnitudes of the same failure mode. The set L's satisfying the above condition can be determined from (J,G), and the probability density for such L's will not have to be calculated, as it will always be zero. More generally, $F(L(k+1))$ is a linear variety in R^{MW} and becomes the empty set only for

certain values of $\underline{L}(k+1)$. Hence the region of integration needs to be determined as a function of $\underline{l}(k+1)$. Although the scalar version of (5-22), where regions are intervals on the real line and are independent of $\underline{L}(k+1)$, has been successfully evaluated using numerical quadrature methods [38], there is presently no efficient method available for solving the general multi-dimensional problem. The difficulties are due to the large dimension of $\underline{L}$ and the complex regions of integration, $S(0,-) \cap F(\underline{L}(k+1))$, as indicated above. However, when the rank of G is the same as the dimension of $\underline{L}$, the region of integration will simplify to $S(0,-)$. This is because when rank G=MW, transitions from and $\underline{L}(k)$ to any $\underline{L}(k+1)$ are possible and $F(\underline{L}(k+1))=R^{MW}$. An algorithm has been developed to perform the integration for this special case in low dimensions, and it will be described in Section 5.2. The condition that rank G = MW is not as restrictive as it appears, and we will see that the algorithm based on this assumption is useful in determining the risks associated with some simplified decisions rules to be discussed in the next two subsections.

In this subsection we have examined the problem of designing sliding window sequential decision rules. Several simplifications directed towards practical solutions have been discussed. In addition, we have identified the structure of the computations required for evaluating the risk and performance associated with a sliding window rule. Based on these insights we will propose two simple decision rules (a simplified sliding window rule and a non-window rule) in Subsections 5.1.3 and 5.1.4 for which the risks can be evaluated or approximated by existing numerical techniques.

5.1.3 A Simplified Sliding Window Decision Rule

The multi-dimensional integrals (5-16)-(5-18) encountered in the calculation of the risk are due to the MW-dimensional vector of decision statistics $L(k)$. These statistics correspond to the MW failure hypotheses, and they provide the information necessary for the simultaneous identification of both failure type and failure time. In most applications, such as the aircraft sensor FDI problem [11] and the detection of freeway traffic incidents [5], the failure time need not be explicitly identified. In such cases, the terminal decision set reduces to $\mathcal{D}^W = \{j: j=1,\ldots,M\}$, where the index j denotes the declaration of a type j failure. Since the decision does not directly concern the onset of failures, the failure time resolution power provided by the full window of decision statistics is not needed. Instead, decision rules that employ a few components of $L(k)$ may be used. The decision rule of this type considered here consists of sequential decision regions that are similar to (5-8) but are only defined in terms of M components of $L(k)$:

$$S_j = \{L_{W-1}(k): L(k;j,W-1) > f_j \quad e^{-1}(j,W-1)[L(k;j,W-1)-f_j] > e^{-1}(i,W-1)[L(k,i,W-1)-f_j],\ \forall i \neq j \qquad (5\text{-}24a)$$

$$S_0 = \{L_{W-1}(k): L(k,j,W-1) \leq f_j \qquad j=1,\ldots,M\} \qquad (5\text{-}24b)$$

where S_j is the stop-to-declare-j- region and S_0 is the continue region. At each point in time, the set of decision regions specified by (5-24) may be regarded as a decision rule for determining if a failure has occurred at the earliest point in the sliding window (i.e. at k-W+1), and it is similar to a W-sample decision rule for testing M+1 hypotheses. This vantage point readily provides a rough guideline for choosing the window size W, namely, W should be sufficiently long so that enough residual samples can be used to achieve acceptable detection accuracy in the non-sequential testing of M+1 hypotheses (M hypotheses indicating the possibility of one of M different failures occurring at k-W+1 plus the hypothesis of no failure at k-W+1). Because the actual decision problem is a sequential one, rather than a static one this guideline will only serve to provide an initial choice of W that may be later adjusted to achieve a better performance. (We will discuss the choice of W along with other design parameters of the Bayesian approach in Subsection 5.3.1). It should be noted that the use of (5-24) is effective if cross-correlations of signatures among hypotheses of the same failure type at different times are smaller than those among hypotheses of different failure types.

Next, we will examine the risk associated with the sequential rule (5-24). The following equations for the risk computation are specializations of those of the previous section to the simplified sliding window rule. We have

$$U_s^W(f) = L_F \sum_{i=1}^{M} \sum_{\tau=W+1}^{\infty} \mu(i,\tau) \sum_{k=W}^{\tau-1} \sum_{j=1}^{M} P_r\{L_{W-1}(k) \in S_j,\ S_0(k-1) \mid 0,-\}$$

$$+ \sum_{i=1}^{M} \sum_{\tau=1}^{\infty} \mu(i,\tau) \sum_{k=\max[W,\tau]}^{\infty} \sum_{j=1}^{M} [c(i)(k-\tau)+L(i,j)]$$

$$\times \Pr\{L_{W-1}(k) \in S_j,\ S_0(k-1) \mid i,\tau\} \qquad (5\text{-}25)$$

where

$$S_0(k) = \{L_{W-1}(k) \in S_0, \ldots, L_{W-1}(W) \in S_0\} \tag{5-26}$$

The probabilities required for calculating the risk (5-25) are given by

$$p(L_{W-1}(k+1) \,|\, S_0(k), i, \tau) = \left[\int_{S_0} p(L_{W-1}(k) \,|\, S_0(k-1), i, \tau)\, dL_{W-1}(k)\right]^{-1}$$

$$\times \int_{S_0} p(L_{W-1}(k+1) \,|\, L_{W-1}(k), S_0(k-1), i, \tau)\, p(L_{W-1}(k) \,|\, S_0(k-1), i, \tau)\, dL_{W-1}(k)$$

$$k>W \tag{5-27}$$

$$\Pr\{L_{W-1}(k) \in S_j,\ S_0(k-1) \,|\, i, \tau\}$$

$$= \Pr\{S_0(k-1) \,|\, i, \tau\} \int_{S_j} p(L_{W-1}(k) \,|\, S_0(k-1), i, \tau)\, dL_{W-1}(k) \quad j=0,1,\ldots,M \tag{5-28}$$

with

$$\Pr\{L_{W-1}(W) \in S_j \,|\, i, \tau\} = \int_{S_j} p(L_{W-1}(W) \,|\, i, \tau)\, dL_{W-1}(W) \tag{5-29}$$

In contrast to the MW-dimensional integrals associated with the sliding window rule discussed in the previous subsection, the integrals in (5-27)-(5-29) are M-dimensional. For M small, say less than 4, numerical integration of (5-27)-(5-29) becomes manageable.

Unfortunately, the transition density, $p(L_{W-1}(k+1) \,|\, L_{W-1}(k), S_0(k-1), i, \tau)$, required in (5-27) is difficult to calculate, because $L_{W-1}(k)$ is not a Markov process. As an alternative, the Markov nature of $L(k)$ can be exploited

once again to determine the required conditional density and probabilities as follows:

$$p(L(k+1)\,|\,S_0(k),i,\tau) = \Big[\int_{\tilde{S}_0} p(L(k)\,|\,S_0(k-1),i,\tau)\,dL(k)\Big]^{-1}$$

$$\times \int_{S_0 \cap F(L(k+1))} |G'G|^{-1} p(r(k+1)=(G'G)^{-1}G'[L(k+1)-JL(k)]\,|\,i,\tau)\,p(L(k)\,|\,S_0(k-1),i,\tau)\,dL(k) \tag{5-30}$$

$$\Pr\{L_{W-1}(k)\in S_j,\ S_0(k-1)\,|\,i,\tau\}$$

$$= \Pr\{S_0(k-1)\,|\,i,\tau\}\int_{\tilde{S}_j} p(L(k)\,|\,S_0(k-1),i,\tau)\,dL(k) \tag{5-31}$$

where $\tilde{S}_j$ is the extension of S_j in the $L(k)$-space, i.e. R^{MW}, i.e. $\tilde{S}_j = \{L(k): L_{W-1}(k)\in S_j\}$, $j=0,1,\ldots,M$. The obvious drawback in using (5-30) and (5-31) is that we have to deal with MW-dimensional integrals again. Therefore, in order to exploit the low dimensionality of (5-27) and (5-28), we will have to use an approximation for the transition density in (5-27). In the remainder of this subsection we will describe a simple approximation of $p(L_{W-1}(k+1)\,|\,L_{W-1}(k),\ S_0(k-1),i,\tau)$.

It is useful to note that in approximating the required transition density for $L_{W-1}(k)$ we are, in fact, approximating the behavior of L_{W-1}. A simple approximation is a Gauss-Markov process $\ell(k)$ that is defined by

$$\ell(k+1) = A\ell(k) + \xi(k+1) \tag{5-32}$$

$$E\{\xi(k)\xi'(t)\} = BB'u_0(k-t) \tag{5-33}$$

where A and B are MxM constant matrices and ξ is a white Gaussian sequence with covariance equal to the (MxM) matrix BB'. The reason for choosing the model (5-32) and (5-33) is twofold. Firstly, just as $L_{W-1}(k)$, $\ell(k)$ is Gaussian. Secondly, $\ell(k)$ is Markov so that its transition density can be readily determined. In order to have $\ell(k)$ behave like $L_{W-1}(k)$, we set the matrices A and B and the mean of ℓ such that

$$E_{i,\tau}\{\ell(k)\} = E_{i,\tau}\{L_{W-1}(k)\} \tag{5-34}$$

$$E_{0,-}\{\ell(k)\ell'(k)\} = E_{0,-}\{L_{W-1}(k)L'_{W-1}(k)\} \tag{5-35}$$

$$E_{0,-}\{\ell(k)\ell'(k+1)\} = E_{0,-}\{L_{W-1}(k)L_{W-1}(k+1)\} \tag{5-36}$$

That is, we have matched the marginal density and the one-step cross-covariance of $\ell(k)$ to those of $L_{W-1}(k)$. It can be shown that (5-34)-(5-36) uniquely specify

$$A = \Sigma_1' \Sigma_0^{-1} \tag{5-37}$$

$$BB' = \Sigma_0 - \Sigma_1' \Sigma_0^{-1} \Sigma_1 \tag{5-38}$$

$$E_{i,\tau}(\xi(k+1) = E_{i,\tau}\{L_{W-1}(k+1)\} - A\, E\{L_{W-1}(k)\} \tag{5-39}$$

where

$$\Sigma_0 = E\{L_{W-1}(k)L'_{W-1}(k)\} = \sum_{t=0}^{W-1} G_t V^{-1} G'_t \tag{5-40}$$

$$\Sigma_1 = E\{L_{W-1}(k)L'_{W-1}(k+1)\} = \sum_{t=0}^{W-2} G_{t-1} V^{-1} G'_t \tag{5-41}$$

$$E_{i,\tau}\{L_{W-1}(k)\} = \begin{cases} 0 & \tau > k \\ \sum_{t=0}^{k-\tau} G_{t-k_0} V^{-1} G'_t & k_0 = k-W+1-\tau \leq 0 \\ \sum_{t=0}^{W-1} G_t V^{-1} G'_{t+k_0} & k_0 = k-W+1-\tau > 0 \end{cases}$$

Moreover, the matrix A is stable, i.e. the magnitudes of all of the eigenvalues of A are less than unity, and B is invertible if G_0 or G_{W-1} is of rank M. Because ξ is an artificial process (i.e. ξ is not a direct function of the residuals r), $\ell(k)$ can never be implemented for use in (5-24).

It should be noted that the model specified by (5-34)-(5-36) does not provide the only Markov approximation of $L_{W-1}(k)$. We may, indeed, choose to match the n-step cross-covariance ($1<n<W$) instead of matching the one-step cross-covariance as in (5-36), or we may just approximate the cross-covariance function. Such a variety of possible models is the result of the relatively small number of free parameters available in the Markov model to be adjusted in order to describe the more complex $L_{W-1}(k)$ process. The suitability of a criterion for choosing the matrices A and B,

such as (5-35) and (5-36), depends directly on the failure signatures under consideration and may be examined as an issue separate from the decision rule design problem. Since our main goal is to demonstrate how the Bayes approach can be used in designing sequential decision rules, we will not pursue this issue further. Rather, we will proceed with the design problem assuming an appropriate Markov approximation (5-32) of $L_{W-1}(k)$ is available.

Now we can approximate the required probabilities in the risk calculation as

$$P_r\{L_{W-1}(k) \in S_j,\ S_0(k-1) \,|\, i,\tau\} \approx P_r\{\ell(k) \in S_j,\ S_0(k-1)\ i,\tau\} \qquad j=0,1,\ldots,M \quad k \geq W \qquad (5\text{-}43)$$

and

$$\Pr\{\ell(k) \in S_j,\ S_0(k-1) \,|\, i,\tau\} = \Pr\{S_0(k-1) \,|\, i,\tau\} \int_{S_j} p(\ell(k) \,|\, S_0(k-1), i,\tau)\, d\ell(k) \qquad (5\text{-}44)$$

where we have applied the same decision rule to $\ell(k)$ as $L_{W-1}(k)$. Therefore, S_j and $S_0(k-1)$ denote the decision regions and the event of continued sampling up to time k for both L_{W-1} and ℓ. Assuming B^{-1} exists, we have

$$p(\ell(k+1) \,|\, S_0(k), i,\tau) = \left[\int_{S_0} p(\ell(k) \,|\, S_0(k-1), i,\tau)\, d\ell(k)\right]^{-1} \times \int_{S_0} p(\xi(k+1) = [\ell(k+1) - A\ell(k)] \,|\, i,\tau)\, p(\ell(k) \,|\, S_0(k-1), i,\tau)\, d\ell(k) \qquad k \geq W \qquad (5\text{-}45)$$

where $p(\xi(k) \,|\, i,\tau)$ is the Gaussian density of $\xi(k)$ under the failure (i,τ).

If $\ell(k)$ satisfies (5-34) and (5-35)

$$\Pr\{\ell(W) \in S_j | i,\tau\} = \Pr\{L_{W-1}(W) \in S_j | i,\tau\} \tag{5-46}$$

In contrast to (5-30) and (5-31), the evaluation of (5-44) and (5-45) requires only M-dimensional integrations over the decision regions. The complication in the region of integration due to $F(\ell(k+1))$ (in 5-30) is absent. An algorithm exploiting existing numerical techniques for computing (5-44)-(5-45) has been developed and it will be discussed in Section 5.2.

In the event that B is not invertible, the region of integration in (5-45) will take the form similar to that of (5-22) in lower dimension (M instead of MW). Such an integral is very difficult to evaluate, and it represents an area for future research. Very often this problem can be circumvented by batch processing the residuals. That is, we may consider the modified residual sequence: $\tilde{r}(\tilde{k}) = [r'(v\tilde{k}-v+1), r'(v\tilde{k}-v+2), \ldots, r'(v\tilde{k})]'$ for some batch size $v>0$ with $\tilde{k}=1,2,\ldots$ as the new time index. In using $\tilde{r}(\tilde{k})$ we have to augment the signatures as: $[g_i'(0), \ldots, g_1'(v-1)]'$, $i=1,\ldots,M$. By a proper choice of v, the rank of G_0 can be increased to M and B will be invertible. An example of the batch process is included in the next subsection. Therefore, we direct our attention to cases where B^{-1} exists. Under this condition, the algorithm in Section 5.2 can be used to obtain approximations of the sequential risk and the detection performance (i.e. the expected decision delays, probability of false alarm, etc.) of the simplified sliding window decision rule. Simulation aimed at assessing the accuracy of the probability approximations (5-43) resulting from the use of the Markov

model $\ell(k)$ described by (5-32)-(5-42) are reported in Chapter 6 together with the actual design of decision regions of the form (5-24) for a two-failure-mode problem.

In concluding this subsection, we note that increased accuracy may be obtained by using a higher order approximation, i.e. when $\ell(k)$ is given by

$$\ell(k) = \bar{C}\,\bar{\ell}(k) \tag{5-47}$$

$$\bar{\ell}(k+1) = \bar{A}\,\bar{\ell}(k) + \bar{B}\,\bar{\xi}(k+1) \tag{5-48}$$

where $\bar{A}$ and $\bar{B}$ are nxn and $\bar{C}$ is Mxn with n>M. The increase in accuracy is achieved at the expense of increased computational complexity, since we have to contend with n-dimensional probability integrals over regions of the form in (5-30). When n=MW, (5-47) and (5-48) will provide an exact description of $L_{W-1}(k)$ and we are, once again, confronted with (5-30) and (5-31). Due to the lack of an efficient algorithm for calculating the required integrals, this subject of higher order approximation is not pursued any further in this thesis.

5.1.4 Non-Window Sequential Decision Rules

In the previous subsection we have discussed the simplified sliding window rule in which the M decision statistics are formed from a window of residual samples. Here we will describe another simple decision rule that has the same decision regions as the simplified sliding window rule

(5-24), but the vector (z) of M decision statistics is obtained differently as follows:

$$z(k+1) = \tilde{A}\, z(k) + \tilde{B}\, r(k+1) \qquad (5\text{-}49)$$

where $\tilde{A}$ is a constant stable MxM matrix, and $\tilde{B}$ is a Mxm constant matrix of rank M. Unlike the Markov model $\ell(k)$ that approximates $\ell_{W-1}(k)$, $z(k)$ is a realizable Markov process driven by the residual. The obvious advantages of using z as the decision statistic are: 1) less storage is required, because residual samples need not be stored as necessary in the sliding window scheme, and 2) since z is Markov, the required probability integrals are of the form (5-44) and (5-45), and the algorithm to be described in Section 5.2 can be directly applied to evaluate such integrals.

In order to form the statistics z, we need to choose the matrices $\tilde{A}$ and $\tilde{B}$. When the failure signatures under consideration are constant biases; $\tilde{B}$ can simply be set to equal G_0, and $\tilde{A}$ can be chosen to be αI, where $0<\alpha<1$. Then, the term Br in (5-49) resembles Gr of (5-19), and it provides the correlation of the residual with the signatures. The time constant $(\frac{1}{1-\alpha})$ of z characterizes the memory span of Z just as W characterizes that of the sliding window rules. When α is close to one, residual samples from long ago are remembered, and when α is zero, Z(k) is just $\tilde{B}r(k)$. Therefore, α (or A in general) can be regarded as a design parameter playing a role similar to that of W in the sliding window scheme.

More generally, if we consider failure signatures that are not constant biases. Then the choice of A may still be handled in the same way as in the

as in the constant-bias case, but the selection of a $\tilde{B}$ matrix is more involved. Qualitatively, the role of $\tilde{B}$ (just as G in (5-19)) is to bring out the failure signature contained in the residual. Therefore the rows of $\tilde{B}$ should represent some characteristic directions of the failure signatures in question, e.g. in the constant-bias case, the rows of $\tilde{B}$ are simply the signatures of the failures. As the signatures are not constants the choice of such characteristic directions is not straightforward and is very much problem dependent. With some insights into the nature of the signatures, a reasonable choice of $\tilde{B}$ can often be made. To illustrate how this may be accomplished, we will consider an example with two failure modes and an m-dimensional residual vector. Let

$$g_1(k-\tau) = \beta_1 \tag{5-50}$$

$$g_2(k-\tau) = \beta_2(k-\tau+1) \tag{5-51}$$

That is, g_1 is a constant bias, and g_2 is a ramp. If β_1 and β_2 are not multiples of each other a simple choice of $\tilde{B}$ is available:

$$\tilde{B} = \begin{bmatrix} \beta_1 \\ \beta_2' \end{bmatrix} \tag{5-52}$$

If $\beta_1=\alpha_1\beta$ and $\beta_2=\alpha_2\beta_2$, where α_1 and α_2 are scalar constants, the above choice of $\tilde{B}$ has rank one and is not very useful for identifying either signature. Suppose we batch process every two residual samples together, i.e we use the residual sequences $\tilde{r}(\tilde{k}) = [r'(2\tilde{k}-1), r'(2\tilde{k})]'$, $\tilde{k}=1,2,\ldots$.

Then we can set $\tilde{B}$ to be

$$\tilde{B} = \begin{bmatrix} \beta' & \beta' \\ \beta' & 2\beta' \end{bmatrix} \tag{5-53}$$

Thus, the first and second rows of $\tilde{B}$ capture the constant-bias and ramp nature of g_1 and g_2, respectively (and this $\tilde{B}$ has rank two). The use of the modified residual $\tilde{r}(\tilde{k})$ in this case causes no adverse effect, since it only lengthens slightly the interval between times when terminal decisions may be made. A big increase in such intervals i.e., the batch processing of $r(k),\ldots,r(k+v)$ simultaneously for large v, may however, be undesirable.

The above simple example serves to show that the applicability of the decision statistic z is not as restrictive as it first appears to be. In any event, the matrices $\tilde{A}$ and $\tilde{B}$ may be regarded as design parameters just as the cost functions and prior probability mass function. The merit of any choice of $\tilde{A}$ and $\tilde{B}$ may be assessed by determining if the decision rule based on such choices yields good performance. The algorithm of Section 5.2 will aid in evaluating the risk associated with using z in the decision rule, and the risk-minimization algorithm to be discussed in Section 5.3 can be used in obtaining a decision rule that has as small a risk as is possible for a particular choice of $\tilde{A}$ and $\tilde{B}$. The design of a decision rule using z as the decision statistic for a two failure-mode problem is reported in Chapter 6.

While the statistic z is potentially useful for a wide range of problems, we expect its effectiveness to diminish for problems where the signatures vary drastically as a function of the elapsed time, or the distinguishability among failures depends eventually on these variations. This is due to the fact that a constant $\tilde{B}$ is not adequate to capture the essence of rapidly varying signatures. In such cases the sliding window decision rule should provide better performance because of its inherent nature to look for a full window's worth of signature. However, this still leaves many applications for which z is a useful statistics.

It is possible to use a higher order z (similar to $\bar{L}$ of the last subsection). It is not considered here, because in fact, it is mimicking the sliding window statistic L_{W-1}. In addition, the increased order complicates both implementation and the computation of the required probability integrals, now having the form (5-30) and (5-31). Such added complexity will negate the advantages of using z.

5.2 Evaluation of the Risk and Performance Probabilities

In this section we will examine the problem of computing the risk associated with the decision rules discussed in the last two subsections. An algorithm based on standard numerical quadrative techniques has been developed for calculating the conditional density (5-45) and probabilities (5-44) recursively. Since the decision statistic z of subsection 5.1.4 and the approximation (ℓ) of the sliding window statistic L_{W-1} of subsection 5.1.3 are both Markov processes with the required calculations in the form

of (5-44) and (5-45), this algorithm is directly applicable to both cases. We will only describe the algorithm for the two failure-mode problem, although it may be easily generalized to an arbitrary number of failure modes. It will become clear, however, that due to the exponential increase in computational requirement as a function of the number of failure modes the algorithm is only practical for decision rules dealing with a few failure modes. This problem is intrinsic to the numerical evaluation of multi-dimensional integrals and, in general, cannot be avoided. The approach to the design of a robust residual generation process undertaken in Chapters 2 and 3 will aid in limiting the number of failure modes to be considered simultaneously by a decision rule. Since each residual generation process is based on a part of the system, namely the most relevant and parameter-insensitive part, it will include only a subset of all the possible failure types of the whole system. Then, it is likely that a decision rule employing the residual from such a process will have to deal with only a small number of failure modes.

A brief review of the quadrature technique employed in this study is included in subsection 5.2.1, while the actual algorithm for calculating the conditional density and the required probabilities is described in subsection 5.2.2. In subsection 5.2.3, we will discuss the risk evaluation problem.

5.2.1 Gaussian Quadrature Formulas

Numerical integration generally involves the approximation of a definite integral by a finite sum. The most widely studied method is of the form

$$\int_{\alpha}^{\beta} w(x)F(x)dx \simeq \sum_{s=1}^{n} v^s F(x^s) \qquad (5\text{-}54)$$

where s is an index for the points used in the formula, x is a scalar variable, and w(x) is a function for which the integrals $\int_{\alpha}^{\beta} w(x)x^k dx$, k=0,1,2,... are defined and finite; v^s and x^s are known as the weights and nodes, respectively. When w(x) is nonnegative in $[\alpha,\beta]$, a set of weights and nodes can be found so that the approximation (5-54) becomes exact for F when it is a polynomial of degree less than 2n. Such an approximation is known as a n-point Gaussian quadrature formula, or Gaussian formula [39]. Based on the theory of orthogonal polynomials, efficient Gaussian formulas have been determined for the 1-dimensional integral for a variety of w and intervals $[\alpha,\beta]$. Attempts to develop similar formulas for several dimensions has met with little success. The most common approach to M-dimensional integration is to regard the integral as a M-fold iterated integral and apply a 1-dimensional formula to each variable separately. The resulting formula is called a product formula, e.g. in two dimensions.

$$\int_{\alpha_2}^{\beta_2}\int_{\alpha_1}^{\beta_1} F(x_1,x_2)dx_1dx_2$$

$$= \int_{\alpha_2}^{\beta_2}\int_{\alpha_1}^{\beta_1} w_1(x_1)w_2(x_2)[w_1^{-1}(x_1)w_2^{-1}(x_2)F(x_1,x_2)]dx_1dx_2$$

$$\simeq \sum_{t=1}^{n_2}\sum_{s=1}^{n_1} v_1^s v_2^t\, w_1^{-1}(x_1^s)w_2^{-1}(x_2^t)F(x_1^s,x_2^t) \qquad (5\text{-}55)$$

where we have assumed $w_\gamma^{-1}(x)$ exists for $\gamma=1,2$, and v_γ^s, x_γ^s are the weights and nodes of the n_γ-point Gaussian formulas:

$$\int_{\alpha_\gamma}^{\beta_\gamma} w_\gamma(x)F(x)dx \simeq \sum_{s=1}^{n_\gamma} v_\gamma^s F(x_\gamma^s), \quad \gamma=1,2, \tag{5-56}$$

This approach is the basis of our algorithm for evaluating the integrals of (5-44) and (5-45). The two 1-dimensional Gaussian quadrature formulas employed in the algorithm are

$$\int_0^\infty e^{-x}F(x) \simeq \sum_{s=1}^{n} \tilde{v}_L^s F(x_L^s) \tag{5-57}$$

$$\int_{-\infty}^{\infty} e^{-x^2}F(x)dx \simeq \sum_{s=1}^{n} \tilde{v}_H^s F(x_H^s) \tag{5-58}$$

Now, $\tilde{v}_L^s$ and x_L^s are the weights and nodes of the n-point Laguerre-Gauss formula (5-57), and $\tilde{v}_H^s$ and x_H^s are the weights and nodes of the n-point Hermite-Gauss formula. The weights and nodes for both of these formulas are tabulated for a wide range of n[40]. (In fact, the nodes x_L^s and x_H^s are the roots of the n-th order Laguerre and Hermite polynomials, respectively.) Provided the integrals exist and are finite, we have the following formulas:

$$\int_0^\infty F(x)dx \simeq \sum_{s=1}^{n} v_L^s F(x_L) \tag{5-59}$$

$$\int_{-\infty}^{\infty} F(x)dx \simeq \sum_{s=1}^{n} v_H^s F(x_H) \tag{5-60}$$

where

$$v_L^s = \tilde{v}_L^s \, e^{x_L^s} \tag{5-61}$$

$$v_H^s = \tilde{v}_H^s \, e^{(x_H^s)^2} \tag{5-62}$$

For some finite regions of integration, such as a sphere, a cube, etc., estimate of the error associated with the product formula (5-55) are available [39]. These results are not useful for our problem, because they are dependent on the (higher order) derivative of the integrand function. The integrands of (5-44) and (5-45) involve the conditional density for which derivative information is very difficult to obtain. The fact that we are dealing with probability integrals, however, will provide us with some handle on the error magnitude. We will discuss this when we describe the algorithm in the next subsection. In closing, we note that further references on numerical integration may be found in the survey paper by Haber [41].

5.2.2 An Algorithm for Calculating the Conditional Density and Associated Probabilities

The computational procedure described here will be applicable to both $\ell(k)$ of subsection 5.1.3 and $z(k)$ of subsection 5.1.4, since by setting $\tilde{Br}(k+1)$ to be $\xi(k+1)$ we can see that both (5-32) and (5-49) have the same form. To facilitate discussion, we will use the simplifying notations:

$$h_k(\ell(k)) = p(\ell(k) \,|\, S_0(k-1), i, \tau) \qquad k>W \tag{5-63}$$

$$h_w(\ell(W)) = p(\ell(W) \,|\, 0, -) \tag{5-64}$$

$$p_k(\ell(k+1)-A\ell(k)) = p(\xi(k+1) = \ell(k+1)-A\ell(k)) \qquad k\geq W \qquad (5\text{-}65)$$

$$P_k(j) = Pr\{\ell(k)\epsilon S_j | S_0(k-1), i, \tau)\} \quad j=0,\ldots,M \qquad k\geq W \qquad (5\text{-}66)$$

where $p(\ell(W))$ is taken to be the steady no-fail density of ℓ, i.e. we assume that we begin in the steady state at W. Note that the dependence on (i,τ) is suppressed. It is understood that the above quantities have to be interpreted in context with some (i,τ) pair.

For M=2, the decision regions have the form (see Figure 5-1):

$$S_0 = \{\ell: \ell_1 \leq f_1, \ell_2 \leq f_2\} \qquad (5\text{-}67a)$$

$$S_1 = \{\ell: e_1^{-1}(\ell_1-f_1) > e_2^{-1}(\ell_2-f_2)\} \qquad (5\text{-}67b)$$

$$S_2 = \{\ell: e_2^{-1}(\ell_2-f_2) > e_1^{-1}(\ell_1-f_1)\} \qquad (5\text{-}67c)$$

Then the propagation of the conditional density is governed by

$$h_{k+1}(\ell(k+1)) = P_k^{-1}(0) \int_{-\infty}^{f_1} \int_{-\infty}^{f_2} p_k(\ell(k+1)-A\ell(k)) h_k(\ell(k)) d\ell_1(k) d\ell_2(k) \qquad (5\text{-}68)$$

Substituting $\ell(k) = f-y$, we get

$$h_{k+1}(\ell(k+1)) = P_k^{-1}(0) \int_0^{\infty} \int_0^{\infty} p_k(\ell(k+1)-A[f-y]) h_k(f-y) dy_1 dy_2 \qquad (5\text{-}69)$$

Similarly, we can write

$$P_k(0) = \int_0^\infty \int_0^\infty h_k(f-y)\,dy_1 dy_2 \tag{5-70}$$

$$P_k(1) = \int_{-\infty}^{\infty} \int_0^\infty h_k\left(\begin{bmatrix} a_1(\ell_2(k))-y_1 \\ \ell_2(k) \end{bmatrix}\right) dy_1 d\ell_2(k) \tag{5-71}$$

$$P_k(2) = \int_{-\infty}^{\infty} \int_0^\infty h_k\left(\begin{bmatrix} \ell_1(k) \\ a_2(\ell_1(k))-y_2 \end{bmatrix}\right) dy_2 d\ell_1(k) \tag{5-72}$$

where

$$a_1(\ell_2(k)) = \begin{cases} f_1 & \ell_2(k) \leq f_2 \\ \dfrac{\epsilon_1}{\epsilon_2}(\ell_2-f_2)+f_1 & \ell_2(k) > f_2 \end{cases} \tag{5-73}$$

$$a_2(\ell_1(k)) = \begin{cases} \dfrac{\epsilon_2}{\epsilon_1}(\ell_1-f_1)+f_2 & \ell_1(k) > f_1 \\ f_2 & \ell_1(k) \leq f_2 \end{cases} \tag{5-74}$$

The integrals (5-69)-(5-72) are in the forms that are suitable for the application of the product formula employing Laguerre and Hermite formulas Using for the integral from 0 to ∞ (5-59) and (5-60) for the integral from $-\infty$ to ∞, the above integrals can be approximated as

$$h_{k+1}(\ell(k+1)) \simeq P_k^{-1}(0) \sum_{t=1}^{n_L} \sum_{s=1}^{n_L} v_L^s v_L^t p_k(\ell(k+1) - A\begin{bmatrix} f_1 - x_L^s \\ f_2 - x_L^t \end{bmatrix}) h_k \left(\begin{bmatrix} f_1 - x_L^s \\ f_2 - x_L^t \end{bmatrix}\right) \tag{5-75}$$

$$P_k(0) \simeq \sum_{t=1}^{n_L} \sum_{s=1}^{n_L} v_L^s v_L^t h_k \left(\begin{bmatrix} f_1 - x_L^s \\ f_2 - x_L^t \end{bmatrix}\right) \tag{5-76}$$

$$P_k(1) \simeq \sum_{t=1}^{n_H} \sum_{s=1}^{n_L} v_L^s v_H^t h_k \left(\begin{bmatrix} a_1(x_H^t) - x_L^s \\ x_H^t \end{bmatrix}\right) \tag{5-77}$$

$$P_k(2) \simeq \sum_{t=1}^{n_H} \sum_{s=1}^{n_L} v_L^s v_H^t h_k \left(\begin{bmatrix} x_H^t \\ a_2(x_H^t) - x_L^s \end{bmatrix}\right) \tag{5-78}$$

where an n_L-point Laguerre formula and an n_H-point Hermite formula are used. The above approximations may be set to be equalities while keeping in mind that the quantities on the left hand sides become approximations of the true ones. Thus, (5-75) and (5-76) describe the propagation of the approximate conditional density. The probabilities of (5-44) can be approximated by using (5-76)-(5-78) in

$$\Pr\{\ell(k) \in S_j, S_0(k-1) | i, \tau\} = P_k(j) \prod_{s=W}^{k-1} P_s(0) \tag{5-79}$$

Due to the fact that only approximate values of $P_k(j)$, $j=0,1,2$ are used in (5-79), the errors may accumate as k increases. Some feel for this

cumulative error is obtained by means of comparisons with simulation results, and it is reported in Chapter 6.

Since the probabilities in (5-76)-(5-78) should sum to one, i.e. $\sum_{j=0}^{2} P_k(j)$ should equal one, $h_k(\ell(k))$ and $P_k(j)$, j=0,1,2 are normalized for all k>W. This insures that $P_k(j)$ are valid probabilities and $h_k(\ell(k))$ will always be close to being a density function. The un-normalized sum, $\sum_{j=0}^{2} P_k(j)$, can serve as a coarse indicator of the accuracy of the approximations. That is, if it is not close to one, we know that the approximations are poor and more points will have to be used in the quadrature. Although the fact that the sum is close to one does not necessarily imply the quadrative is indeed accurate, we would be more confident in the approximations if this is in fact the case.

Upon examining (5-75)-(5-78), we note that for every k, there are only a fixed number of points in the ℓ-plane for which h_k will have to determined, namely, n_L^2 points in S_0 and $n_L x n_H$ point in S_1 and S_2 each. Moreover, these points do not vary with k. In order to calculate h_{k+1} for any point ℓ, all n_L^2 points in S_0 will have to be used. An estimate of the computational requirement can be obtained as follows. For simplicity, let us assume $n_L=n_H=n$. Suppose we call the evaluation of each term in the summation in (5-75) a step. Then we need to perform n^2 steps to obtain a new points. Since there are a total of $3n^2$ points (due to three decision regions), $3n^4$ steps will have to be carried out at each iteration. Table 5-1 shows the

n	$3n^4$ (steps/iter)	time/iter (sec)
8	12288	.61
10	30000	1.50
12	62208	3.11
14	115248	5.76
16	196608	9.83
18	314928	15.74
20	480000	24.00
22	702768	35.14
24	995328	49.77
26	1370928	68.55
28	1843968	92.20
30	2430000	121.50

Assuming 5×10^{-5} sec/step

TABLE 5-1: Computation Requirements

total number of steps and the corresponding computation time for iteration as a function of n. (For this table we have assumed a time of 5×10^{-5}/sec/step, while using an un-optimized code on an IBM 370/168 computer we have experienced roughly 7×10^{-5} sec/step.) For n=16, say, almost 10 seconds are required per iteration. Typically, a minimum of 30 iterations are required to calculate the risk approximately (see Section 5.2.3 for a discussion of risk evaluation). This gives 5 minutes per risk evaluation. In searching for a set of risk-minimizing decision regions (see Section 5.3.1), quite a few risk evaluations are usually needed. Therefore, even for the simple two-failure mode case, the computational burden is not insignificant. In general, the M-mode problem will require $(M+1)n^M$ steps/iteration. It is obvious that M does not have to be very large before the amount of computation becomes too much to handle.

In order to construct a Laguerre or Hermite formula that gives good results we need to choose two design parameters carefully - the number of points to be used in the quadrative and the scale factor of the variable of integration. Although we will use the Laguerre formula to illustrate the importance of these two parameters, the following discussion will apply to the Hermite case as well as to product formulas composed of Hermite and Laguerre quadrature rules. Figure 5-2 shows the location of the nodes of a 8-point and a 16-point Laguerre formula. It is evident that for a larger n, the nodes cover a larger range as well as providing a more dense covering for small values of x. Thus, a large n is suitable for integrands that are "spread-out". When more points are used, more computation is required. The choice of an appropriate n is based on the tradeoff between accuracy and computational

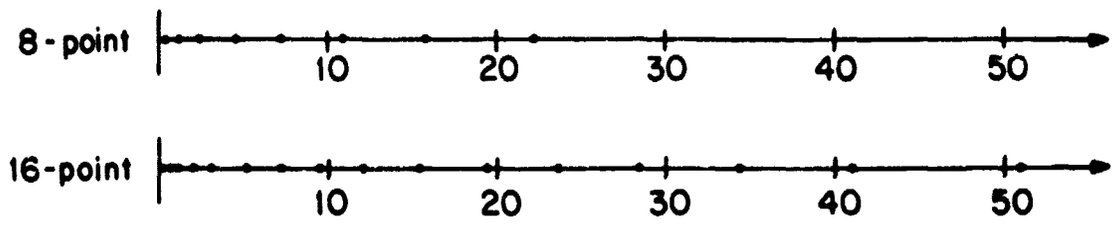


FIGURE 5-2: Nodes of Laguerre Formulas.

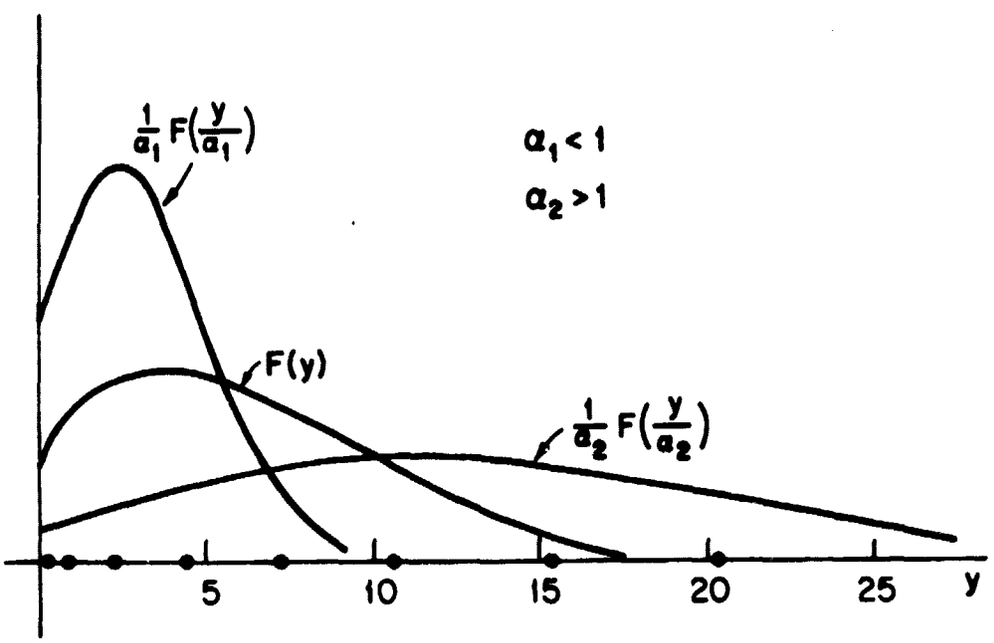


FIGURE 5-3: Effects of Different Scale Factors.

load. The importance of an appropriate scaling of the integrand can be illustrated by considering a simple 1-dimensional integral $\int_0^\infty F(x)dx$. With a simple scale change of the variable of integration, we can obtain $\int_0^\infty F(x)dx = \int_0^\infty \frac{1}{\alpha} F(\frac{y}{\alpha})dy$, where $y=\alpha x$ and α is positive. The situation of applying the same Laguerre formula to the integral with different scale factors α is shown in Figure 5-3. The dots on the y-axis mark the nodes. A small α has the effect of compressing the integrand while a large α tends to spread it out. For an excessively large α, the range of the nodes does not span the integrand sufficiently. On the other hand, for an excessively small α, the nodes do not capture sufficient details of F. In both of these cases, the approximation of the integral is expected to be poor. A good choice of α can only be made with some insights into the nature of the integrand.

For the present problem, these two parameters are chosen heuristically according to the above guidelines. Important considerations include the spread of the integrands of (5-75)-(5-78) and the thresholds (f). The spreads of the integrands of interest are generally difficult to calculate. Consider the integrand of (5-75). It is the product of p_k and h_k. Therefore, its spread is roughly the spread of the minimum of the spreads of p_k and h_w. Since the covariance of p_k is much smaller than that of h_w, the scaling of the variable of integration is chosen to minimize the adverse effect (of Figure 5-3) on p_k. For simplicity, the same scaling is used for all integrals, although different ones may be applied for more accurate results. Algorithms

using different scaling for each of the integrands in (5-75)-(5-78) should be investigated in the future.

Depending on their sizes, the thresholds may also be the source of error for the quadrative. To see this, let us consider the 1-dimensional analog of (5-75), i.e. integrating over the continue region, as pictured in Figure 5-4. According to the formula (5-75), the nodes of the Leguerre formula are located relative to the threshold. For the larger threshold f^1, many of the nodes span the insignificant part of the integrand, while for f^2, the nodes of the Laguerre formula (with the same number of points) cover the integrand well. (Note that scaling will not improve the situation). Ideally, n, the number of points in the quadrature formula, should be chosen as large as is practical so that sufficient number of nodes will cover the significant part of the integrand. In this study n is roughly chosen so that the span of the nodes (the distance between the minimum and maximum nodes) is a few times the sum of the magnitude of the maximum threshold and the scaled spread (the square root of the largest diagonal element of the covariance matrix) of h_w. As the spread of h_w is an approximation of that of h_k, this choice of n will provide sufficient covering for the integrands of (5-76)-(5-78) as well as that of (5-75). (The spread of the former is larger than that of the latter).

It is noted that the span of the Hermite nodes are very small even for n_H large (see the table of Hermite roots and weights in [40]). The Hermite formula is used in computing the stopping probabilities $P_k(1)$ and $P_k(2)$. In (5-77) and (5-78), the Hermite nodes are centered at the axes. Under the

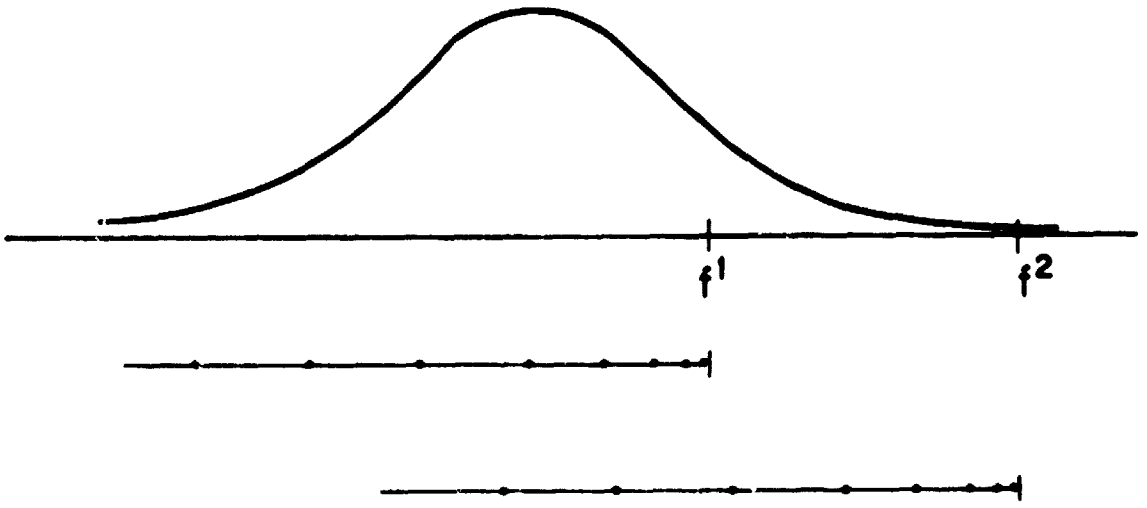


FIGURE 5-4: Effect of the Threshold.

no-fail hypothesis this will provide a sufficient covering of p_k in S_1 and S_2. Under a failure nypothesis, p_k will shift into the stopping region. Thus, the short span of the Hermite roots together with the centering at the axes may not cover p_k in S_1 and S_2 well enough. To compensate for this, we will shift the Hermite nodes into S_1 and S_2. This is accomplished by modifying (5-77) and (5-78) as

$$p_k(1) \simeq \sum_{t=1}^{n_H} \sum_{s=1}^{n_L} v_L^s v_H^t h_k \left(\begin{bmatrix} a_1(x_H^t) - x_L^s \\ x_H^t + \lambda_1(k;i,\bar{\tau}) \end{bmatrix} \right) \qquad (5\text{-}77)'$$

$$p_k(2) \simeq \sum_{t=1}^{n_H} \sum_{s=1}^{n_L} v_L^s v_H^t h_k \left(\begin{bmatrix} x_H^t + \lambda_2(k;i,\bar{\tau}) \\ a_2(x_H^t) - x_L^s \end{bmatrix} \right) \qquad (5\text{-}78)'$$

where

$$\lambda_j(k;i,\tau) = \min[f_{3-j}, E(\ell_{3-j}(k) \mid i,\bar{\tau})], \quad j=1,2$$

That is, as p_k shifts into S_1 and S_2 (under a failure), the Hermite nodes are also shifted into S_1 and S_2 by means of λ_1 and λ_2. Note that λ_1 and λ_2 are clipped at the threshold values. This prevents the Hermite nodes to move too far away from the thresholds into the stopping regions in the events that the expected value of $\ell(k)$ under a failure grows indefinitely. Thus, we have constructed a moving grid to cover the conditional density p_k in order to obtain better accuracy in the quadrature. (Comparisons among the results from using (5-77) and (5-78), (5-77)' and (5-78)', and Monte Carlo

simulations have shown the moving-grid approach to be more accurate that the static method). The technique of using a moving grid should be further studied in future efforts in developing algorithms for computing the detection probabilities.

In summary, we have described an algorithm for calculating (approximately) the conditional density and the probabilities required for the risk evaluation; and we have provided an assessment of its practicality. Effective choice of design parameters, the scale factor of the variable of integration and the number of points used in the Laguerre and Hermite formulas, has been discussed. The performance is assessed via comparisons with Monte Carlo simulations for various types of signatures and thresholds, and the result is reported in Chapter 6. In addition to aiding the design of decision rules, the present algorithm provides a simple framework for exploring finer issues of computing the probabilities so that the design of more effective and efficient algorithms can be facilitated. Finally, we note that integration algorithms based on other 1-dimensional formulas [39] may be constructed for our problem, although they are not examined in this study.

5.2.3 Risk and Performance Probabilities

In this section we will discuss the computation of the risk and performance probabilities for a sequential decision rule in the form of (5-67) for the detection and identification of the various failure modes (but not failure time). Before we proceed with the calculation, we will examine the behavior of the conditional density $p(\ell(k)|S_0(k-1),i,\tau)$ as a

function of k. Recall that ℓ is described by

$$\ell(k+1) = A\ell(k) + \xi(k+1) \tag{5-32}$$

where A is a constant stable matrix, and ξ is a white Gaussian sequence with constant covariance BB'. In the absence of a failure, ξ is zero mean, and thus, is a stationary process. Based on the fact that A is constant, ξ is stationary and the decision regions (in particular, the continue region) are time independent, we conjecture that the conditional density under (0,-) will approach a limiting density function as time progresses, i.e.

$$\lim_{k\to\infty} p(\ell(k) \mid S_0(k-1),0-) = \bar{p}(\ell(k)) \tag{5-80}$$

Qualitatively, the propagation of the conditional density consists of the following process. After the density at k outside the continue region is set to zero, it is normalized to become the density at k conditioned or continuing. Then it is compressed (by the effect of A with eigenvalues of magnitudes smaller than one) and convolved with the density of ξ. The convolution has the effect of spreading the density out again over into the continue region. Since the matrix A, the density of ξ, and the continue region are all time-invariant, a steady state density is most likely to be reached. In fact, convergence is evident in all propagations of the conditional density, by means of the algorithm described in the last subsection, for various values of A, BB' and f (the thresholds defining S_j). Following similar reasoning, we also conjecture that the actual conditional density of

the sliding window rule, i.e. $p(L_{W-1}(k)|S_0(k-1),0-)$, behaves similarly. Unfortunately, we have not been able to prove such convergence behavior using elementary techniques. More advanced function-theoretic methods may be necessary, but they are beyond the scope of this thesis. Assuming that this behavior holds, however, we will be able to obtain a simple approximate expression for the sequential risk. We will discuss this next.

Assuming no failure has occurred, the conditional density will essentially reach a steady state at some finite time T>W. Then, for $k \geq T$ we have

$$\Pr\{\ell(k) \in S_j | S_0(k-1),0-\} = \bar{b}_j \tag{5-81}$$

$$\Pr\{\ell(k) \in S_j, \ell(k-1) \in S_0, \ldots, \ell(\tau) \in S_0 | S(\tau-1), i, \tau\} = b_j(k-\tau|i) \quad k \geq \tau \geq T \tag{5-82}$$

That is, once steady state is reached, only the relative time (elapsed time) is important. Generally, failures occur infrequently, and decision rule with low false alarm probabilities are employed. Thus, it is reasonable to assume 1) $\rho \ll 1$ ($(1-\rho)^T \simeq 1$), i.e. the failure rate is low (see Sec. 4.2), and 2) $\Pr\{S_0(T)|0,-\} \simeq 1$, i.e. nearly no false alarm before steady state is reached. Using (4-3), the sequential risk (5-25) for M=2 can be approximated by

$$\begin{aligned} U_s^W(f) &\simeq L_F \sum_{i=1}^{2} \sum_{\tau=T+1}^{\infty} \sigma(i)\rho(1-\rho)^{\tau-1} \sum_{k=T}^{\tau-1} \sum_{j=1}^{2} [\bar{b}_j \bar{b}_0^{k-T} \Pr\{S_0(T)|0,-\}] \\ &\quad + \sum_{i=1}^{2} \sum_{\tau=T+1}^{\infty} \sigma(i)\rho(1-\rho)^{\tau-1} \sum_{k=\tau}^{\infty} \sum_{j=1}^{2} [c(i)(k-\tau)+L(i,j)] b_j(k-\tau) \bar{b}_0^{k-T} \Pr\{S_0(T)|0,-\} \\ &= \left[\frac{(1-\rho)(1-\bar{b}_0)}{1-\bar{b}_0(1-\rho)}\right] L_F + \left[\frac{\rho}{1-\bar{b}_0(1-\rho)}\right] \sum_{i=1}^{2} \sigma(i) \sum_{j=1}^{2} \sum_{t=0}^{\infty} [c(i)t+L(i,j)] b_j(t|i) \end{aligned} \tag{5-83}$$

Next, we will seek to replace the infinite sum over t in (5-83) by the finite sum up to t=Δ plus a term approximating the remainder of the infinite sum. Suppose we have been sampling for Δ steps since the failure occurred. Recall the notation (5-66):

$$P_t(j) = \Pr\{\ell(t) \in S_j | S_0(t-1), i, 0\} \quad j=0,1,2 \tag{5-84}$$

If we stop computing the probabilities after Δ, we may approximate

$$P_t(j) \simeq P_\Delta(j) \qquad j=0,1,2, \quad t>\Delta \tag{5-85}$$

and consequently

$$b_j(t|i) \simeq b_0(\Delta|i)\ P_\Delta^{t-\Delta}(0)\ P_\Delta(j) \qquad t>\Delta \tag{5-86}$$

Under the no-fail hypothesis, (5-81) implies that (5-85) is good for $\Delta \geq T$. When the signature of the failure modei is a constant, the same reasoning behind (5-81) may be applied, and we can see that $P_t(j)$ under failure mode i will reach a steady state value as t (the elapsed time) increases. In this case, (5-85) is also a valid approximation for a large Δ. Generally, the failure signatures of interest are not necessarily constant. However, for sufficiently large Δ, the probability of continuing after Δ time steps (since the failure occurred) may be arbitrarily small. The error introduced by (5-85) in the risk (and performance probability) calculation is, consequently, small. Thus, we see that the approximation (5-85) is a reasonable one for a sufficiently large Δ.

Substituting (5-86) in (5-83), we get

$$U_s^W(f) \simeq P_F L_F + (1-P_F) \sum_{i=1}^{2} \sigma(i) \left[a(i)\bar{t}_i + \sum_{j=1}^{2} L(i,j) P(i,j) \right] \tag{5-87}$$

where

$$P_F = \frac{(1-\rho)(1-\bar{b}_0)}{1-\bar{b}_0(1-\rho)} \tag{5-88}$$

$$\bar{t}_i = \sum_{j=1}^{2} \sum_{t=0}^{\Delta} t\, b_j(t|i) + b_0(\Delta \mid i)\, \Delta + \frac{1}{1-P_\Delta(0)} \tag{5-89}$$

$$P(i,j) = \sum_{t=0}^{\Delta} b_j(t|i) + b_0(\Delta|i) \frac{P_\Delta(j)}{1-P_\Delta(0)} \tag{5-90}$$

P_j is the unconditional false alarm probability, i.e. the probability of one false alarm over all time, $\bar{t}_i$ is the conditional expected delay to decision, given that a type i failure has occurred, and P(i,j) is the conditional probability of declaring a type j failure, given that failure i has occurred. From the assumption that $\Pr\{S_0(T) \mid 0,-\} \approx 1$ and the steady condition (5-81), it can be shown that the mean time between false alarms is simply $(1-\bar{b}_0)^{-1}$. Now all the probabilities in (5-88)-(5-90) can be computed by using the algorithm of Subsection 5.2.2. Note that the risk expression (5-87) consists only of finite sums. In contrast to the original risk expression (5-25) for the simplified sliding window rule, (5-87) can be evaluated with a reasonable amount of computational effort. With such an

approximation of the sequential risk, we will be able to consider the problem of determining the decision regions (the thresholds f) that minimize the risk. We will discuss the risk-minimization problem in the next section.

It should be noted that we are not limited to only consider the risk as the objective function in the decision rule design problem. For example, we could consider choosing a set of thresholds that minimize a weighted combination of certain detection probabilities (P(i,j)), the expected detection delay ($\bar{t}_1$), and the mean time between false alarms ($(1-\bar{b}_0)^{-1}$). Although such an objective function will not result in a Bayesian design in general, it is a valid design criterion that may be useful for some application. Since these non-Bayesian objective functions are also functions of the performance indices (expected delay, etc.), they can be evaluated using the approach described in this subsection and the previous one. Although we will not directly consider the non-Bayesian design problems, the risk-minization algorithm and the choice of design parameters discussed in the next subsection are also applicable for these problems.

5.3 Design of Decision Rule - Choice of Design Parameters and Minimization of the Risk

For a given set of cost functions, prior PMF, and other design parameters, such as the window length W, and the matrices $\tilde{A}$ and $\tilde{B}$ used in forming the decision statistic z, the design of a suboptimal rule essentially amounts to determining a set of decision regions (characterized by the thresholds f)

which minimizes the sequential risk. An algorithm, which is especially suitable for this minimization problem, is described in subsection 5.3.1. The effectiveness of the resulting decision rule depends heavily on the choice of the above mentioned design parameters. For example, an improperly chosen cost function that overly penalizes false alarms will result in prolonged decision delays under failures. A window that is too short may not utilize sufficient data to achieve good detection performance regardless of any choice of cost functions. This aspect of the decision rule design problem will be discussed in subsection 5.3.2.

5.3.1 The Sequence-of-Quadratic-Programs (SQP) Algorithm for Minimizing the Risk

The risk minimization problem has two features that deserve special attention. Firstly, the sequential risk is not a simple function of the threshold f, and the derivative with respect to f is not readily available. Secondly, calculating the risk is a costly task. Therefore, the minimum-seeking procedure to be used must require few function (risk) evaluations, and it must not require derivatives. The sequence-of-quadratic-programs (SQP) algorithm studied by Winfield [42] has been chosen to solve this problem, because it does not need any derivative information and it appears to require fewer function evaluations than other well-known algorithms [42]. Furthermore, the SQP is simple, and it has quadratic convergence. We will describe the SQP for the 2-dimensional case, but the generalization to higher dimensions is straightforward.

Applications of the SQP to the risk minimization involves iterating through the following steps:

1) Initially, six different sets of thresholds are picked, and the corresponding sequential risks are calculated. The threshold set having the smallest risk is called the base point (denoted by f^0), and the remaining five sets are indexed according to increasing distance from f^0, i.e. f^1 is the closest and f^5 is the farthest from f^0.

2) A quadratic function described by

$$u(x) = \frac{1}{2} x'Hx + c'x + U_s^W(f^0) \tag{5-91}$$

where

$$x = f - f^0 \tag{5-92}$$

H is a 2x2 symmetric matrix, and c is a 2-vector, is fitted through the six threshold-risk pairs (with the base point as the origin). That is, H and c are determined from the equations

$$U_s^W(f^j) - U_s^W(f^0) = \frac{1}{2} (f^j-f^0)'H(f^j-f^0) + c'(f^j-f^0), \quad j=1,\ldots,5 \tag{5-93}$$

Note that H does not have to be positive definite. The quadratic function u approximates the risk in a region spanned by $\{f^0,\ldots,f^5\}$.

3) The constraint region, R, is defined to be the square region centered at the base point with sides parallel to the axes. It is over R that the minimization of u(x) will be performed in Step 4. The length of the sides, γ, is given by

$$\gamma = 2 \times .99 \times \frac{\epsilon}{\sqrt{2}} \kappa \tag{5-94}$$

where ϵ is the distance between f^0 and f^5, and γ is limited by some appropriate γ_{max}, which has the connotation as the maximum step size; K is a constraint region reduction factor that is initially set to one but as we will see, it may be modified in subsequent steps according to the outcome of the minimization effort at each step. In other words, R is a square inscribed in a circle centered at f^0 with radius .99 ϵ K.

4) The quadratic function u(x) obtained in Step 2 is minimized over the constraint region - this is a quadratic program. The square R, in fact, has been specified to make the quadratic programming solution procedure simple. Let x^M denote the solution. From (5-92), x^M corresponds to a threshold set $f^M = x^M + f^0$. Since u(x) is quadratic, a solution lying in the interior of R has to be a global minimum. Therefore, x^M is such a solution if i) x^M is in the interior of R, ii) H is positive definite, and iii) grad $u(x^M)=0$. Otherwise, the solution lies on the sides of the square. Along each side of the square one component of X is fixed and u(x) is a quadratic function of the remaining free component. Therefore, this is a 1-dimensional analog of the previous condition, and similar reasoning can be applied to determine if a minimum of the 1-dimensional quadratic lies on the side but not at the corners. There may be a maximum of four such minima. The smallest of these will be the solution. If no such minimum exists, the four corners of the square will be examined. The corner giving the smallest u will be the solution.

5) If $U_s^W(f^M) < U_s^W(f^0)$, f^M is used as the new base point (and re-labelled as f^0). Five points that are closest to the new base point are selected and

labelled as $f^1,\ldots,f^5$ according to increasing distance from f^0. The constraint region reduction faster K is set to one, and the procedure is repeated starting at Step 2.

6) If $U_s^W(f^M) < U_s^W(f^0)$, the old base point is kept. Five points closest to f^0 are selected and labelled as $f^1,\ldots,f^5$ according to increasing distance from f^0. Since R excludes the old f^5 (see (5-94)), f^M will always be closer to f^0 than the old f^5 is, and f^M will be included in the five new point. This is the mechanism that provides the algorithm with the learning from mistakes. The reduction factor κ is set to be smaller than one (say .95), so as to limit the searching a little closer to f^0. The procedure is repeated at Step 2.

As convergence is approached, the minimum of the quadratic program will occur inside the square. The procedure is terminated when it is evident that a local minimum has been approached. This algorithm prescribes a sequence of quadratic programs, hence the name SQP. Finally, we note that in theory all the thresholds and the associated risks may be kept so that they will be available as candidates for the 5 closest points in Steps 5 or 6. In practice, however, it is sufficient to store only a few more in addition to the active six.

5.3.2 The Choice of Design Parameters

There are basically two types of design parameters in the present methodology: those affecting the information content of the decision statistics and those that play the role of weights in the risk expression.

The former type includes the window length W and the matrices $\tilde{A}$ and $\tilde{B}$ of subsection 5.1.4. The latter type includes the cost functions and the prior PMF. We will discuss them separately.

The window length determines how much data is to be used in forming the decision statistics. If the signatures do not vanish, an increased window length will improve the signal to noise ratio, and hence will also improve detection performance. For the simplified sliding window rule (that uses only $L_{W-1}(k)$), a long window will cause the decision statistics to remember the past too well so that they become sluggish in responding to failures. However, such an increase in detection delay is absent when a full window of decision statistics (i.e. the complete $L(k)$) is used as in the original sliding window rule (5-8).

Viewing the decision problem as a W-sample, non-sequential problem, as we have mentioned previously, may cast some light on how to choose W for the simplified sliding window rule. From this view point, the choice of W becomes determining how many samples should be used. A more simplified situation may be obtained by considering each of the M failure modes separately, i.e. we now have M W-sample binary hypotheses testing problems at hand. Then, it is clear that W should be chosen such that it is not excessively long but still give a high signal (signature) to noise ratio for each mode. In addition, W must be large enough so that all failure signatures over the window are sufficiently different from one another. A reasonable choice for the window length is some value somewhat greater than all the W's for the binary hypotheses testing problems. With some assumed value of probabilities of false alarm and detection for the binary hypotheses test, reasonable choices of thresholds

can be made. These thresholds may, in turn, be used as initial guesses of the thresholds required for the SQP algorithm for the risk-minimization. A refinement of the choice of W can always be made after evaluating the performance probabilities of the window rule using the initial choice.

Recall that the matrix $\tilde{A}$ used in forming the Markov decision statistics z also plays the role of a memory parameter, and it may be chosen with the same considerations. We will let $\tilde{A}$ be a bit more general here than in subsection 5.1.4., i.e. $\tilde{A}$ is now a diagonal matrix with elements between 0 and 1, and the diagonal elements of $\tilde{A}$ may be different from one another. A diagonal $\tilde{A}$ is used to provide a separate memory for each component of z. Consider the i-th component of z

$$z_i(k+1) = \alpha_i z_i(k) + b_i' r(k+1) \qquad (5\text{-}95)$$

where α_i is the ith diagonal element of $\tilde{A}$ and b_i' is the ith row of $\tilde{B}$. The signal to noise ratio of z is its mean under the ith failure divided by its steady state standard deviation. Since an α_i close to one gives z a longer memory, α_i should be chosen so that it is not extremely close to one and that the signal to noise ratio reaches a good level in a reasonable time i.e. not sluggish (the same issue as using L_{W-1}). For choosing $\tilde{B}$, however, there is not a simple general guideline as we have pointed out in subsection 5.1.4. We may, for example, employ the batch processing techniques and take the augmented vector $[g_i'(0),\ldots,g'(v)]$ (where v is the batch size) to be the i-th row of $\tilde{B}$ as in the example of subsection 5.1.4. Generally, each individual case will have to be examined separately. Just as in the choice of W, the

values of $\tilde{A}$ and $\tilde{B}$ may be adjusted should the detection performance be not acceptable.

The prior PMF may be chosen according to the reliability of components, for instance. In such cases the inverse of the Bernoulli parameter has the significance as the mean time to failure. In other cases it may be regarded as a design parameter that is used to aid in specifying the tradeoff between false alarm and other (expected delay and cross-detection) costs.

The cost functions are chosen to reflect how undesirable false alarms, delays in detection, and incorrect detections are relative to one another. Unlike W, $\tilde{A}$, and $\tilde{B}$, which affect the information content of the decision statistics and the risk in a complex manner, the cost functions enters the risk linearly. Hence a change in the cost functions can be accommodated easily without having to re-compute all the performance indices (false alarm probabilities, conditional expected delays in decision, and conditional incorrect detection probabilities), provided they have been stored. In order to arrive at an acceptable design, very often a few sets of cost functions may have to be tried.

5.4 Summary

In this chapter we have described a Bayesian methodology for designing sequential decision rules for FDI. We have examined in detail the three step of the design process: 1) the definition of the decision rule structure, 2) the evaluation of the sequential risk and detection performance, and 3) the choice of design parameters and risk-minimization.

The suboptimal rules studied are time-invariant rules that partition the sample space of the decision statistics into decision regions. The two major types of decision rule examined are the sliding window rule and those that use the decision statistics z. The computational requirement for determining different forms of these decision rules has been assessed. A numerical algorithm based on 1-dimensional quadrature formulas was developed to provide an approximate evaluation of the risk associated with two simple sequential decision rules. The SQP algorithm has been chosen to determine the set of thresholds that minimizes the risk. Finally, the issues involved in choosing the design parameters such as the cost functions, prior PMF, window size W, and the matrices $\tilde{A}$ and $\tilde{B}$ used to generate the statistic z, were discussed.

CHAPTER 6

SEQUENTIAL DECISION RULE DESIGN - A NUMERICAL EXAMPLE

6.1 Introduction

In Chapter 5 we described a methodology for designing sequential decision rules for FDI. Here, we will apply this design approach to a numerical example in order to gain some insights into the nature of this methodology.

In the previous chapter we discussed the design of simplified sliding window rules, i.e. decision rules that use the log likelihood ratios (L_{w-1}) corresponding to the earliest point in the window (Section 5.1.3). Because the computation of probabilities associated with such decision rules is very difficult, we proposed to approximate L_{w-1} by a Markov process ℓ for the purpose of design and performance analysis only, as this does not represent an implementable algorithm (see Section 5.1.3). A quadrature algorithm based on Gaussian quadrature formulas was developed for computing the probabilities associated with decision rules using Markov statistics. (Section 5.2.2). Thus, this algorithm can be used to calculate the probabilities, and hence, the risk associated with the statistic ℓ. Such a risk provides an approximation to the risk associated with a sliding window rule (using L_{w-1}). As a practical alternative, we proposed the use of an implementable Markov statistic z in place of L_{w-1} in the decision rule (Section 5.1.4). The advantages of using such rules are that the statistic z is easier to compute than L_{w-1} and the quadrature algorithm can be applied directly (since z is Markov). Finally, the design of decision rules was formulated as the choice of a set of thresholds that minimizes the risk, and the SQP algorithm (Section 5.3.1) was proposed as a means for performing the risk minimization.

Through an exercise of these design concepts and simulation studies, we can 1) assess the accuracy of the quadrature algorithm for computing (detection) probabilities, 2) determine if the approximation of the sliding window decision statistic L_{w-1} by the Markov statistic ℓ is reasonable, 3) gain some experience with the SQP algorithm for minimizing the risk function, and 4) compare the performance of decision rules using the sliding window statistic L_{w-1} with that using the simpler Markov statistic z. These are the goals of studying a numerical example. We will describe the set-up of the numerical problem in Section 6.2, and we will discuss the results in Section 6.3.

To facilitate discussions, we will introduce the following terminology. We will refer to a simulation of the sliding window rule by SW, a simulation of the rule using the Markov statistic z as Markov implementation (MI), and a simulation of the nonimplementable decision process using the approximation ℓ as Markov approximation (MA).

6.2 The Numerical Example

In the numerical example, we will consider the detection and identification of two possible failure modes (without identifying the failure times). We assume that the residual is a 2-dimensional vector, and the vector failure signatures, $g_i(t)$, $i=1,2$, as functions of the elapsed time t are shown in Table 6-1. The signature of the first failure mode is simply a constant vector. The first component of $g_2(t)$ is a constant, while the second

component is a ramp. We have chosen to examine these two types of signature behavior (constant bias and ramp) because they are simple and describe a large variety of failure signatures that are commonly seen in practice (see, for example [9]). A constant bias represents a constant failure effect on the residuals and such a signature often occurs in practice (e.g. in the detection of biases in sensors). Also, constant signatures can be used to approximate a slowly changing signature, while a ramp can be used to model failure effects that become more noticeable as time progresses. For simplicity, we have chosen V, the covariance of r, to be the identity matrix.

We will design both a simplified sliding window rule (that uses L_{W-1}) and a rule using the Markov statistic z. In the remainder of this section we will discuss the choice of design parameters such as W, L, etc.

$$g_1(t) = \begin{bmatrix} 1 \\ .5 \end{bmatrix}$$

$$g_2(t) = \begin{bmatrix} .5 \\ .25+ \quad .25t \end{bmatrix}$$

$$V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

TABLE 6-1: Failure signatures.

Recall in Section 5.3 that the window size W of a sliding window rule should be chosen so that a reasonably high signature-to-noise ratio (denoted by η_i) for each failure mode is attained. That is, η_i is the ratio between the expected value of $L_{w-1}(k)$, given that a type i failure has occurred at k-W+1, and the standard deviation of the i-th component of L_{w-1}. From the expressions for L_{w-1} given in (5-9)(5-10), it is easy to see that

$$\eta_i = \Sigma_0^{\frac{1}{2}}(i,i) \tag{6-1}$$

where $\Sigma_0(i,i)$ is the (i,i) element of Σ_0, the covariance of L_{w-1}. For our problem, an η of better than 3 can be attained for each failure mode by using a window size of 8, and we will consider simplified sliding window rules with W=8. The approximation (ℓ) of L_{w-1} is given by (5-32)

$$\ell(k+1) = A\,\ell(k) + B\,\xi(k)$$

The values of A, BB', and Σ_0 can be directly determined using (5-33)-(5-41), and they are shown in Table 6-2.

The Markov statistic z is given by (5-49)

$$z(k+1) = \tilde{A}\,z(k) + \tilde{B}\,r(k+1)$$

In order to achieve roughly the same memory span for L_{w-1} and z for this problem, we have chosen $\tilde{A}$ to be a diagonal matrix with both diagonal elements equal to .875 (which roughly gives a time constant of 8 steps for z). The first row of $\tilde{B}$ is set to be [1,.5], i.e. g_1', because to detect a

type 1 failure we have to look for g_1 in the residual. The first component of g_2 is a constant (=.5). Therefore, the (2,1) element of $\tilde{B}$ is set to be .5. The second component of g_2 grows with the elapsed time. In order to exploit this behavior the (2,2) element of $\tilde{B}$ has to be a relatively large number. In this example, we choose it to be 2, the value of the second component of g_2 at a elapsed time of 7*. The values of $\tilde{A}$, $\tilde{B}$, and the steady state covariance (Σ_z) of z are summarized in Table 6.3.

Recall from Section 5.1.3, the decision regions we have chosen take the form (5-24)

$$S_j = \{L_{w-1}(k):$$

$$L(k;j,W-1) > f_j$$

$$\varepsilon^{-1}(j,W-1)[L(k;j,W-1)-f_j] > \varepsilon^{-1}(i,W-1)[L(k;i,W-1)-f_i], \quad i \neq j\}$$

$$S_0 = \{L_{W-1}(k): L(k;j,W-1) \leq f_j, \ j=1,2\}$$

where $\varepsilon(j,W-1)$ is the standard deviation of the j-th component of the statistic L_{W-1}, i.e. $\varepsilon(j,W-1) = \Sigma_0^{1/2}(j,j)$. For the decision rule using z, we only need to substitute z for L_{W-1} and the standard deviation $\Sigma_z^{1/2}(j,j)$ for $\varepsilon(j,W-1)$. Using the data contained in Table 6-2 and 6-3,

* By choosing a large value for this component we are, in some sense looking for a large bias. This means that in using the resulting static we will have difficulty in detecting this failure at or shortly after the onset, because the ramp component will be small. A good choice of $\tilde{B}$ will depend on the signatures of all the failure modes examined as a whole as well as the performance tradeoff prescribed by the use functions. As we have pointed out in Chapter 5 the design of $\tilde{A}$ and $\tilde{B}$ represents an interesting open problem.

$$W = 8$$

$$A = \begin{bmatrix} .826 & .058 \\ .116 & .837 \end{bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} 10 & 8.5 \\ 8.5 & 14.75 \end{bmatrix}$$

$$BB' = \begin{bmatrix} 2.32 & 2.01 \\ 2.01 & 4.58 \end{bmatrix}$$

TABLE 6-2: Parameters for L_{W-1} and ℓ.

$$\tilde{A} = \begin{bmatrix} .875 & 0 \\ 0 & .875 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} 1 & .5 \\ .5 & 2 \end{bmatrix}$$

$$\Sigma_z = \begin{bmatrix} 5.33 & 6.40 \\ 6.40 & 18.13 \end{bmatrix}$$

$$\tilde{B}V\tilde{B}' = \begin{bmatrix} 1.25 & 1.50 \\ 1.50 & 4.25 \end{bmatrix}$$

TABLE 6-3: Parameters for z.

the decision regions corresponding to rules using L_{W-1} and $\tilde{s}$ are depicted in Figures 6-1 and 6-2 respectively. The decision rule design problem consists of choosing a set of thresholds (f_1 and f_2) that minimizes the risk.

The cost functions and prior probability mass function used in this example are shown in Table 6-4. All incorrect identification of failures modes are penalized with 10 units, while correct failure mode identifications are not penalized. False alarm cost is 9 units-false alarms and incorrect failure mode identifications are nearly equally undesirable. The delay cost for both failure modes is chosen to be one. Both failure modes are assumed to be equally likely, and the mean time to failure is 5000 steps. The prior probability μ is fixed by this a priori information. Although it is not done in this study, we note that if the decision rules resulting from the present values of the parameters L, C, μ, W, $\tilde{A}$, and $\tilde{B}$ are unsatisfactory, these parameters may be adjusted to get a new design.

Recall that the risk is an infinite sum over t, the elapsed time (5-83). In Section 5.2.3 we proposed to approximate it as a finite sum (i.e. the original infinite sum truncated at $t=\Delta$) plus a remainder term. For all decision thresholds considered in the present example, the value of Δ is chosen to be 8, which is large enough so that the remainder term is small, but small enough so that the computational load (due to the propagation of the conditional density) remain manageable.

The steady state conditional density, given that no failure has occurred and no false alarm has been declared, is approximated by the

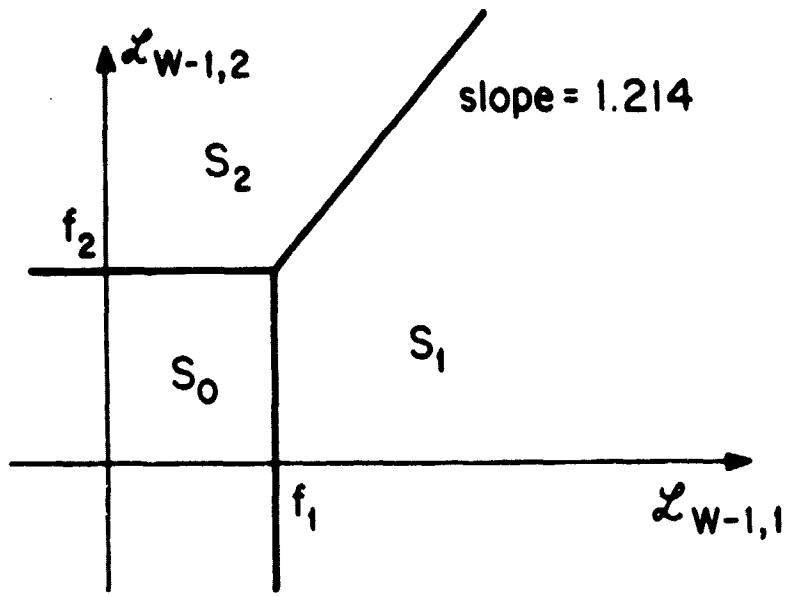


FIGURE 6-1: Decision Regions for Sliding Window Rule.

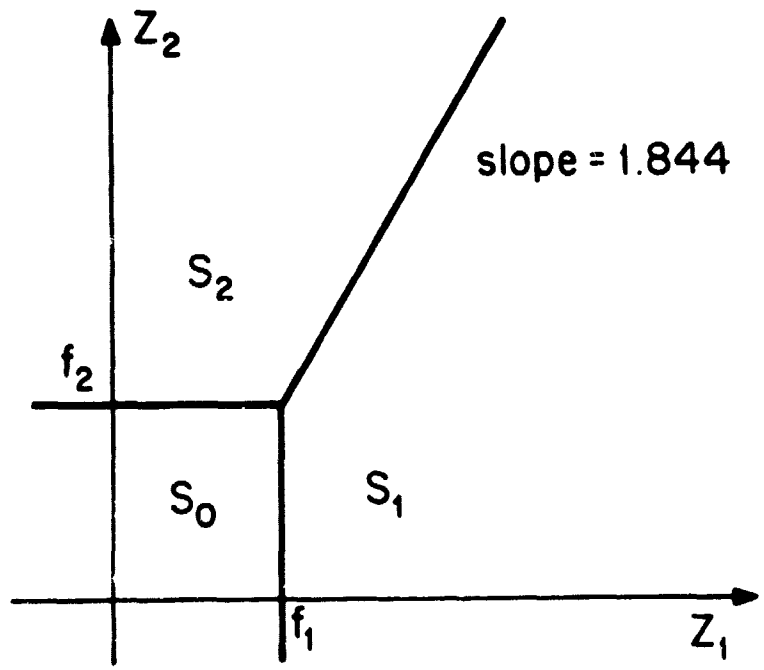


FIGURE 6-2: Decision Regions for a Rule Using z.

$$L_F = 9$$

$$L(1,2) = L(2,1) = 10$$

$$L(1,1) = L(2,2) = 0$$

$$C_1 = C_2 = 1$$

$$\mu(i,\tau) = .5\rho(1-\rho)^{\tau-1},\ i=1,2$$

$$\rho = .0002$$

TABLE 6-4: Cost Functions and Prior Probability.

conditional density propagated 8 steps. That is, we assume T=8 (see Section 5.2.3). Experimentation with larger values of T indicated that the detection probabilities are not significantly changed and this approximation is a reasonable one. To maintain consistency, T is kept at 8 for all thresholds examined.

In order to obtain accurate results, we have to use as large an n_L and n_H (the number of points in the Laguerre and Hermite formulas) as is practical in the quadrature algorithm. In designing decision rules for this example, we will use $n_L = n_H = 20$. With Δ=8 and T=8, the evaluation of the risk for each threshold pair takes approximately 6 minutes of CPU time on IBM370/168 computer.

Recall (Section 5.3.1) that the six threshold pairs required to start the SQP algorithm may be chosen arbitrarily. No set rule is the best for all applications. Here, we will arbitrarily choose these thresholds to take on values within a range of threshold values. They are limited to be positive and within 2 to 4 standard deviations of the decision statistics.

Next, we will discuss the results of applying our design method to this example.

6.3 Results and Discussion

In this section, we will describe our experience with the decision rule design methodology through its application to the example introduced in the preceding section. We indicated that there are four main aspects of the design approach (see Section 6.1) that need to be examined. We will discuss them in the following.

Accuracy of the quadrature algorithm

The quadrature algorithm for computing probabilities is based on Markov decision statistics. In order to determine its accuracy, we have to compare the probabilities associated with a decision rule using Markov statistics (e.g. z) as computed by this algorithm to those obtained via Monte Carlo simulation of the same decision process.

A convenient set of probabilities to be examined include $b_j(t|i)$ (5-82) and $\beta_j(t|i)$ defined by

$$b_j(t|i) = \Pr\{z(t)\epsilon S_j,\ z(t-1)\epsilon S_0,\ldots,z(0)\epsilon S_0|i\},\quad i,j=0,1,2 \qquad (6\text{-}2)$$

$$\beta_j(t|i) = \sum_{s=0}^{t} b_j(s|i),\quad i=0,1,2,\quad j=1,2, \qquad (6\text{-}3)$$

That is, $b_j(t|i)$ is the probability of continuing sampling up to elapsed time t and choosing decision j at t, given i is the true failure mode, and $\beta_j(t|i)$ is the probability of stopping to declare a j-th failure at or before elapsed time t, given i is the true failure mode. Another indicator of the accuracy of the quadrature algorithm as discussed in Section 5.2.2 is P_T:

$$P_T = \sum_{j=0}^{2} P_t(j) \qquad (6\text{-}4)$$

where

$$P_t(j) = \Pr\{z(t)\epsilon S_j|z(t-1)\epsilon S_0,\ldots,z(0)\epsilon S_0,i\} \qquad (6\text{-}5)$$

$P_t(j)$ is the conditional probability of choosing j at t, given we have been sampling through elapsed time t-1 and the true failure mode is i. Therefore, if the algorithm is valid, P_T will be close to 1.

In Figures 6-3 to 6-7 we have shown $b_0(t|0)$, $b_0(t|1)$, $b_0(t|2)$, $\beta_1(t|1)$, and $\beta_2(t|2)$, respectively for a decision rule using the statistic z (as described in Section 6.2) with thresholds f = [6.287, 11.867]'. Plotted against the elapsed time t, $b_0(t|0)$ shows the failure characteristics of the decision rule - a slowly decreasing $b_0(t|0)$ indicates that a low false alarm rate is achieved. The rate of decrease of $b_0(t|i)$, i=1,2, indicates the speed of response of the decision process to failures, and $\beta_i(t|i)$, i=1,2, shows the ability of the decision rule to identify the failure correctly.

The results obtained via Monte Carlo simulation using 10,000 trajectories are marked as MI (see terminology defined in Section 6.1). The quadrature results using $n_L = n_H = 20$ are marked as Q20. The quadrature results using $n_L = n_H = 14$ is also shown in the above figures and they are marked as Q14.

Generally, the quadrature results Q20 are quite close to the simulation results MI, while the quadrature results Q14 are not. This shows that the quadrature algorithm may be used to provide a reasonable approximation of the probabilities, and the accuracy of the approximation can be improved by increasing the number of grid points used in the algorithm. The value of P_T for Q20 ranges from .998 to 1.05, while that for Q14 ranges from .994 to 1.2. Therefore, the value of P_T also indicates that Q20 will provide a close estimate of MI and that Q14 will not.

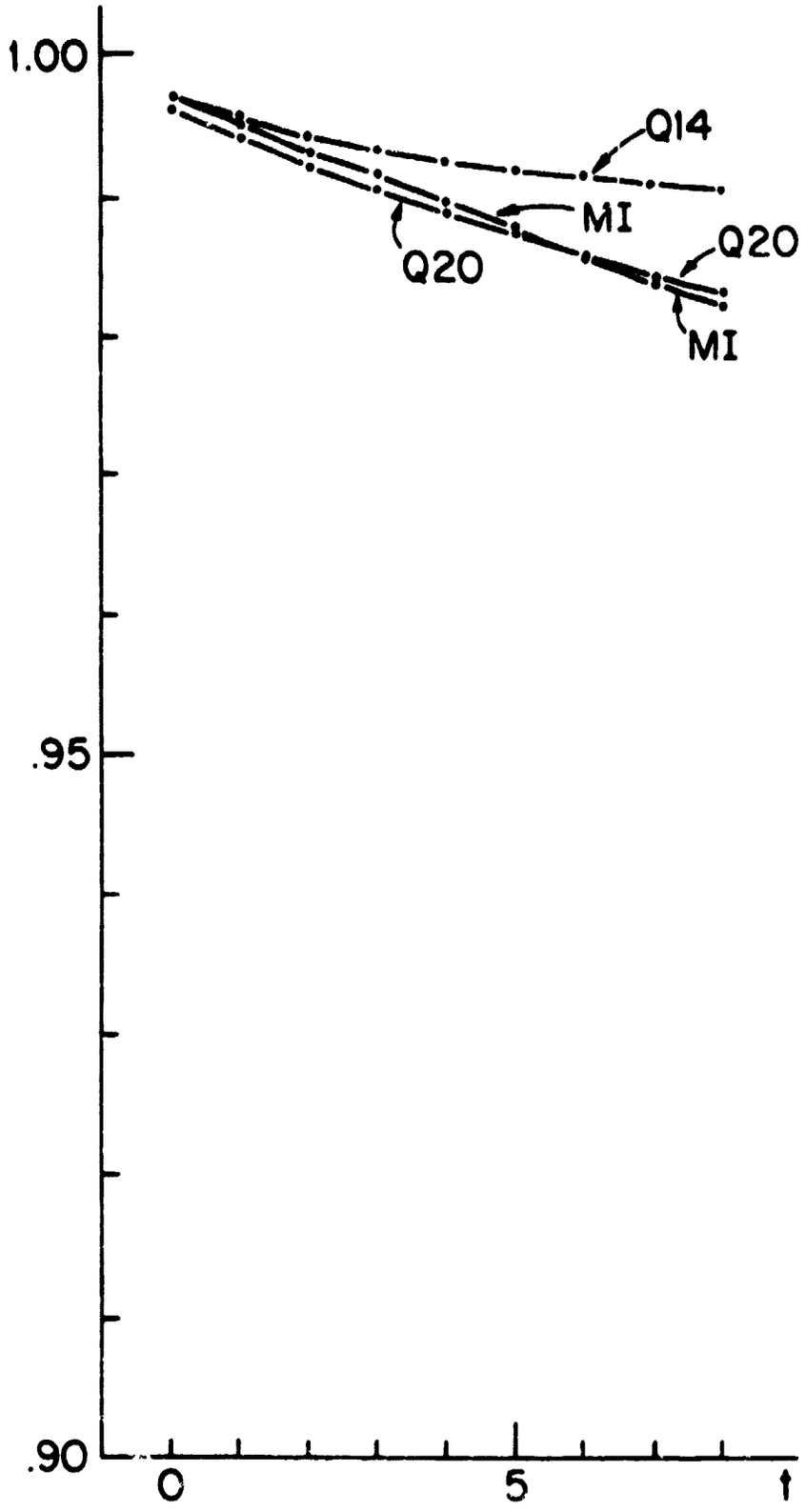


FIGURE 6-3: $b_0(t|0)$ - Using z.

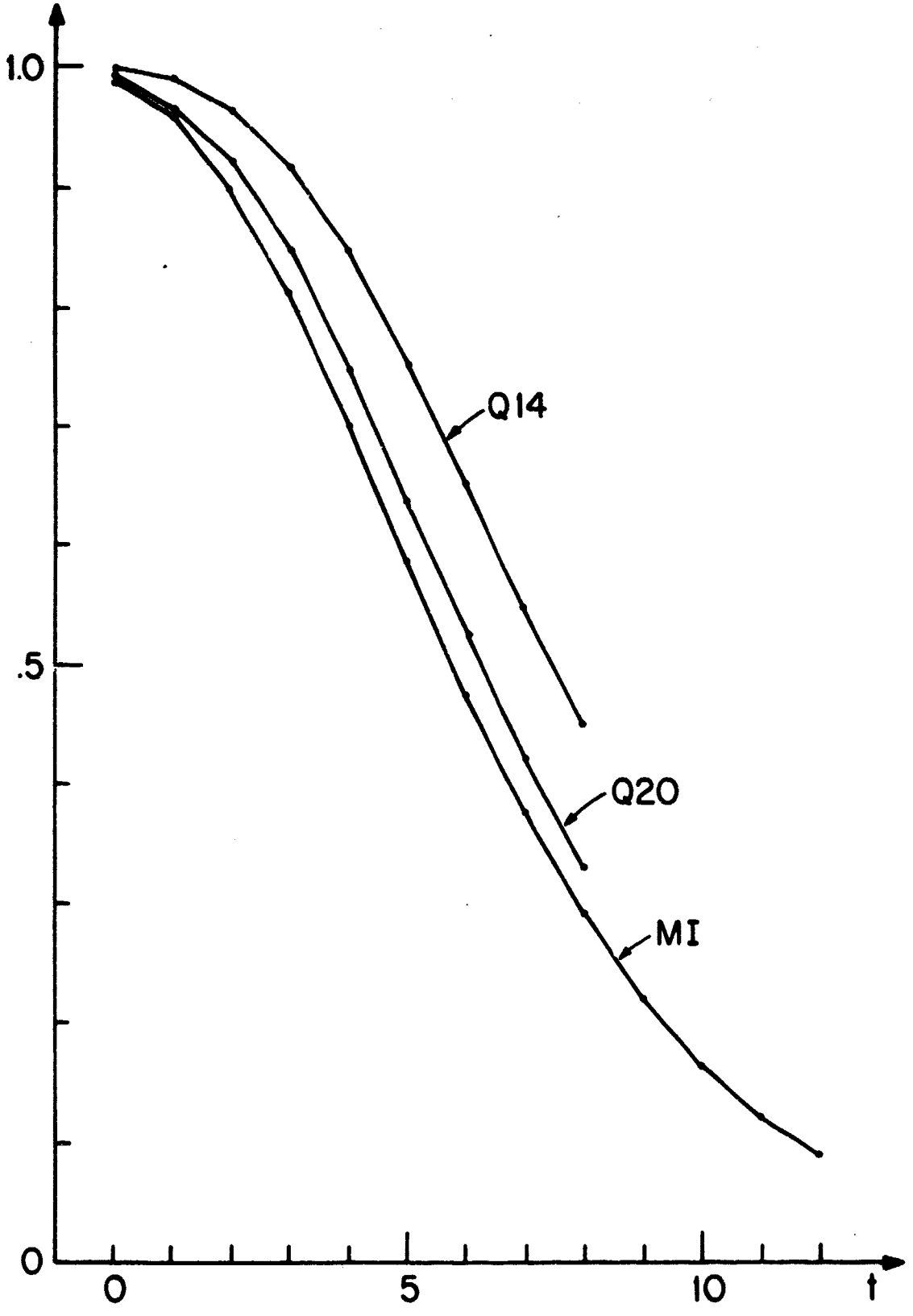


FIGURE 6-4: $b_0(t|1)$ - Using z.

C-3

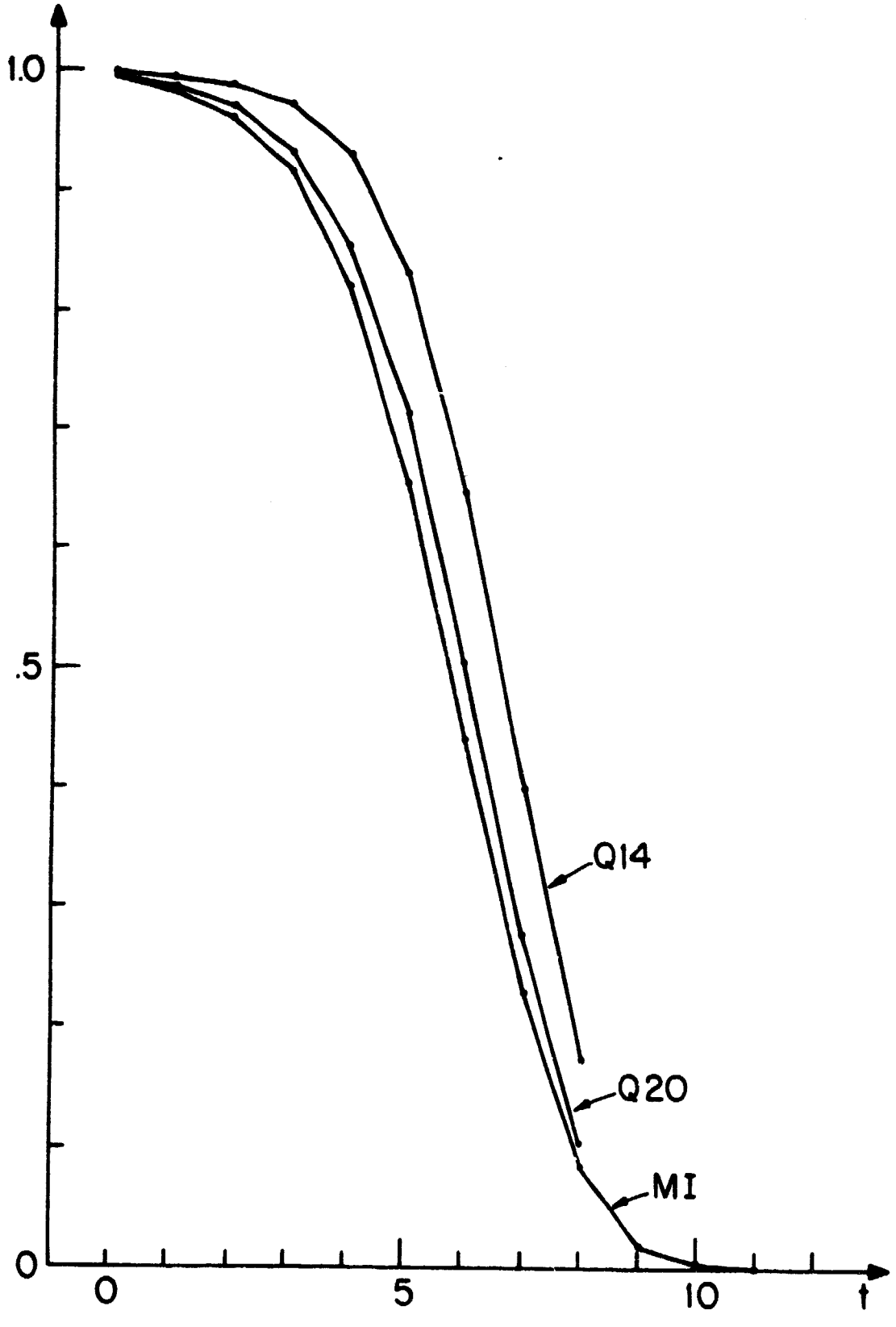


FIGURE 6-5: $b_0(t|2)$ - Using z.

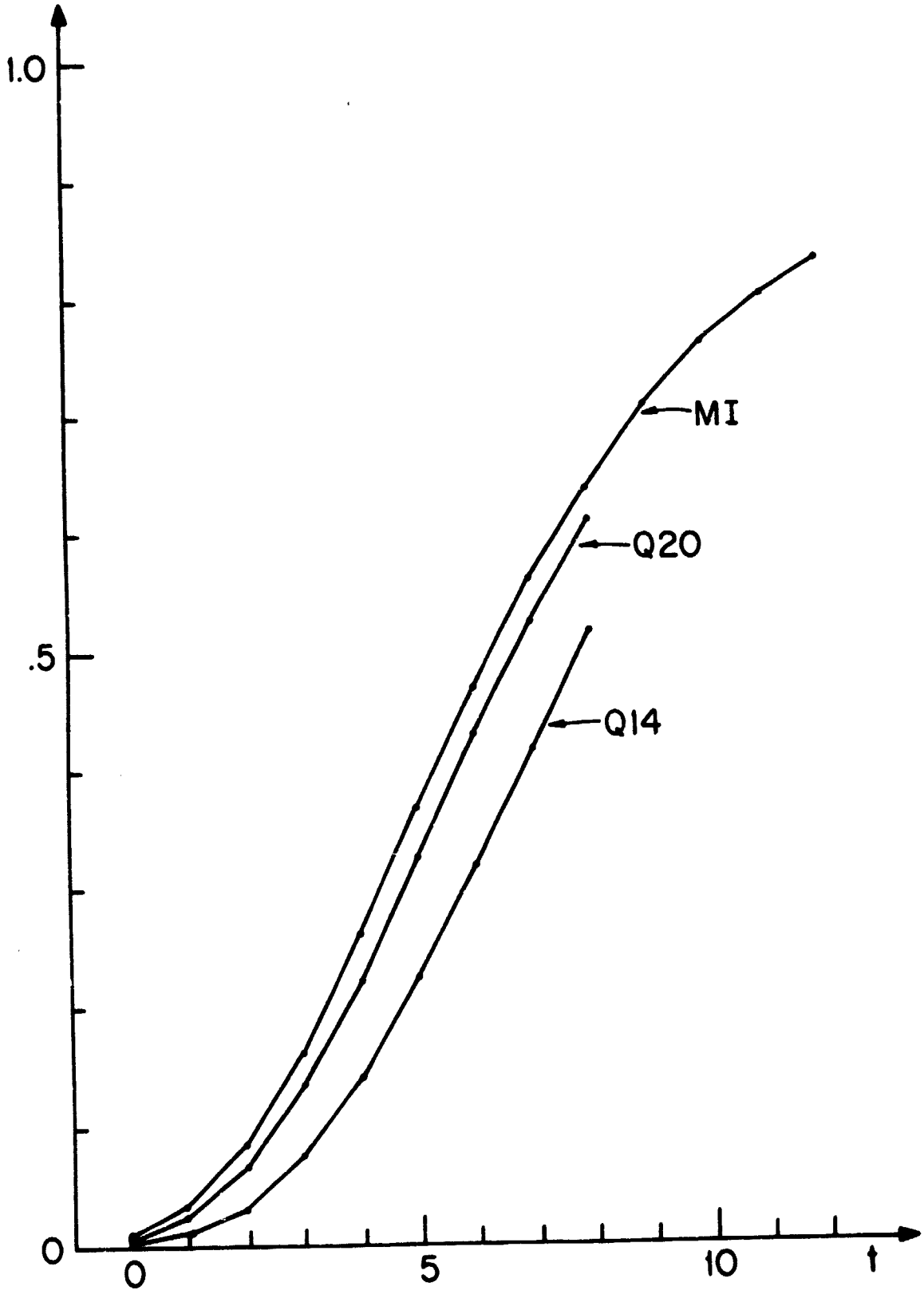


FIGURE 6-6: $\beta_1(t|1)$ - Using z.

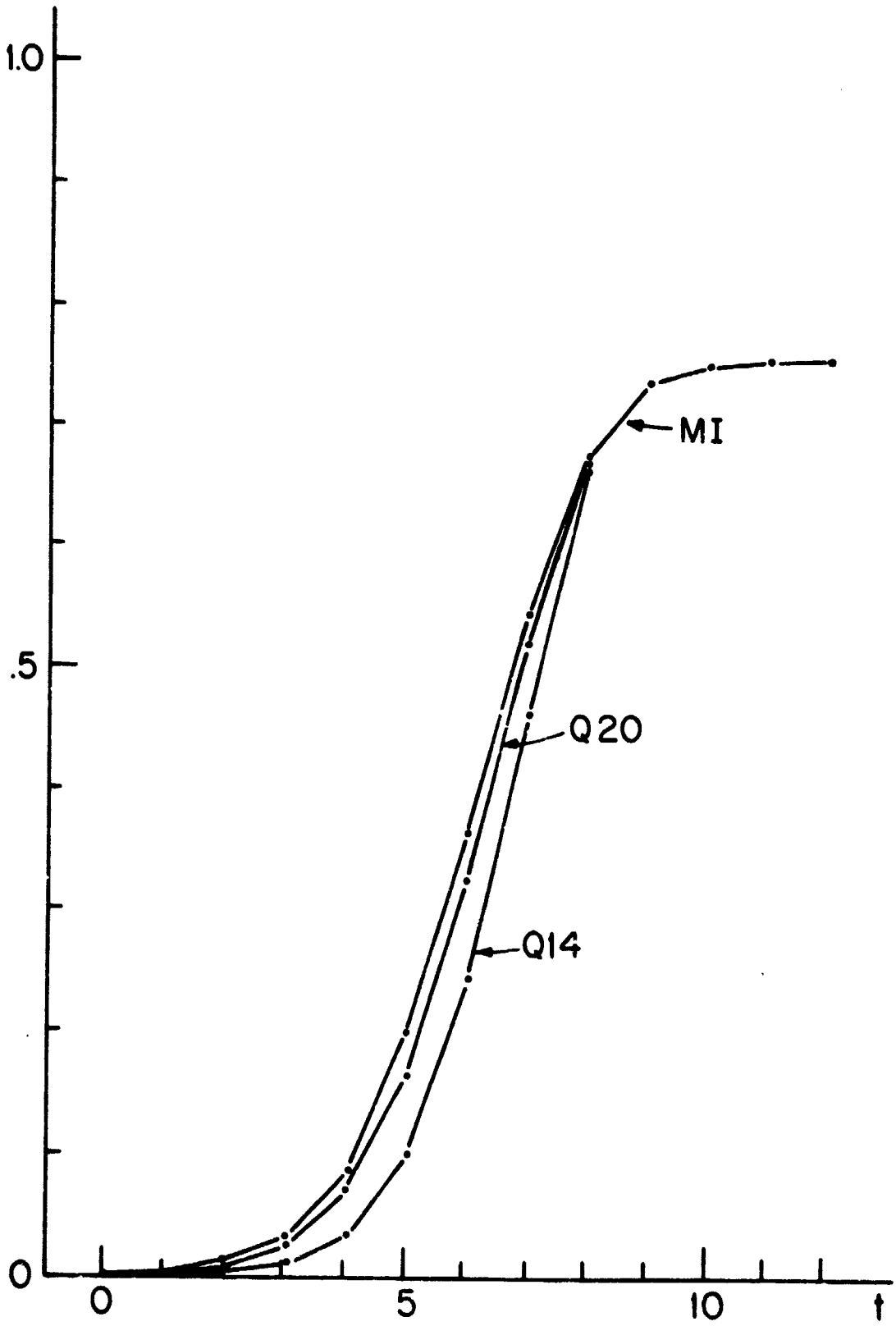


FIGURE 6-7: $\beta_2(t|2)$ - Using z.

Now we compare Q20 with MI. From the trend of $b_0(t|\theta)$ shown in Figure 6-3, it is evident that Q20 will under-estimate the false alarm rate of the decision rule. Both $b_0(t|1)$ and $b_0(t|2)$ indicate that the quadrature results under-estimate the speed of detection. Q20 also slightly under-estimates the correct detection probabilities $\beta_i(t|i)$. On the whole Q20 has provided a reasonably good estimate of the true probabilities associated with the decision rule using the Markov statistic z and threshold [6.287, 11.867].

Finally we note that if we use the quadrature calculations for comparisons between different rules (e.g. for optimization purpose, or just to assess the effect of increasing W, adding a sensor, etc.), then small quadrature errorswill probably have a small effect on the result. When the quadrature errors are consistently in the same direction, e.g. if the quadrature approximation consistently under-estimates false alarm probability and over-estimates detection delay, the relative performance of two decision rules (i.e. which is better than the other) will probably be correctly determined by the approximation method.

The Markov approximation ℓ

Here we will describe a conclusion drawn from a simulation (SW) of the sliding window rule with $f = [8.849, 12.047]'$ and a simulation (MA) of the nonimplementable decision rule based on ℓ and using the same thresholds. (Each simulation consists of 10,000 trajectories.) The resulting probabilities are shown in Figures 6-8 to 6-12. In addition, the probabilities associated with the Markov approximation (ℓ) are

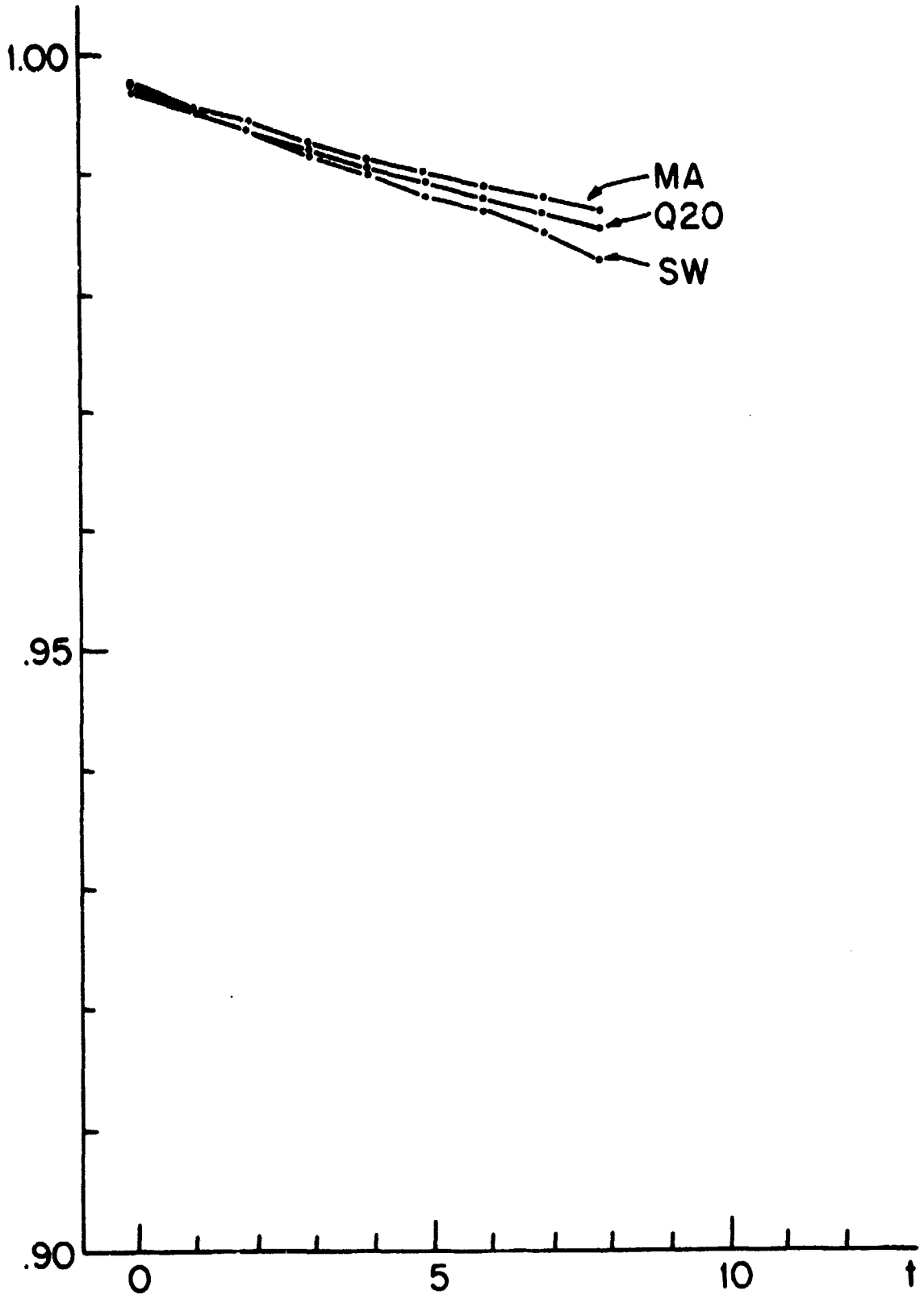


FIGURE 6-8: $b_0(t|0)$ - Sliding Window Rule and Approximation.

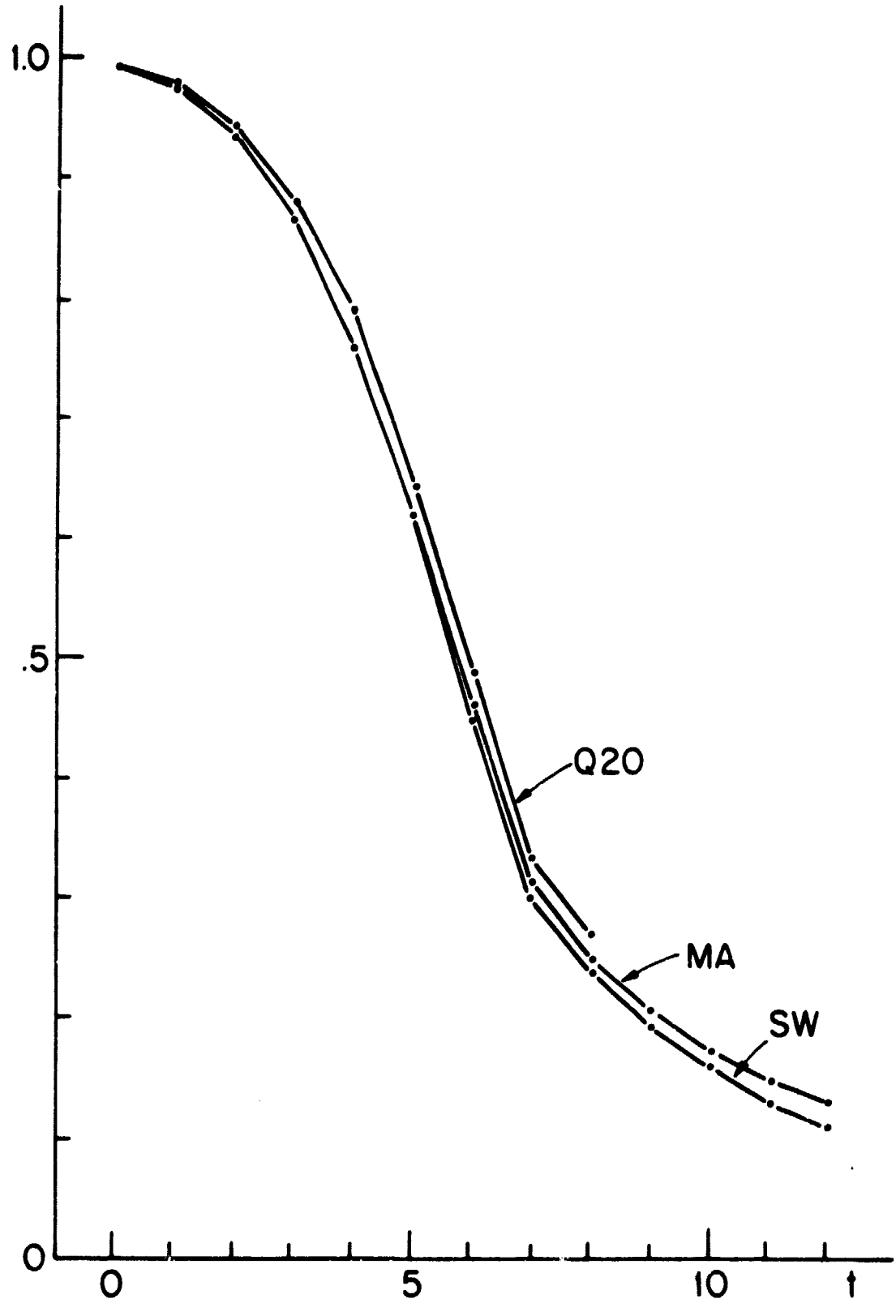


FIGURE 6-9: $b_j(t|1)$ - Sliding Window Rule and Approximation.

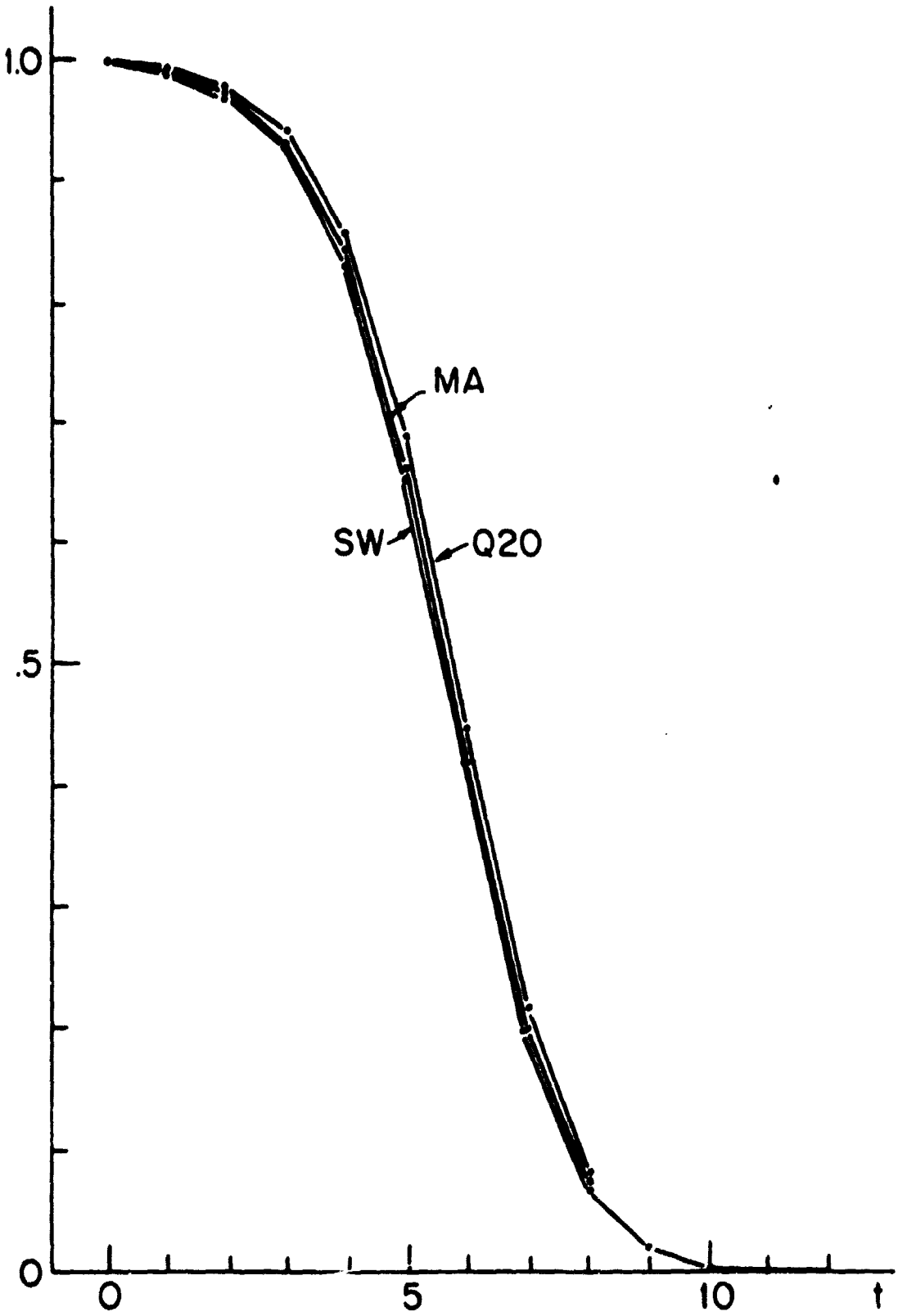


FIGURE 6-10: $b_0(t|2)$ - Sliding Window Rule and Approximation.

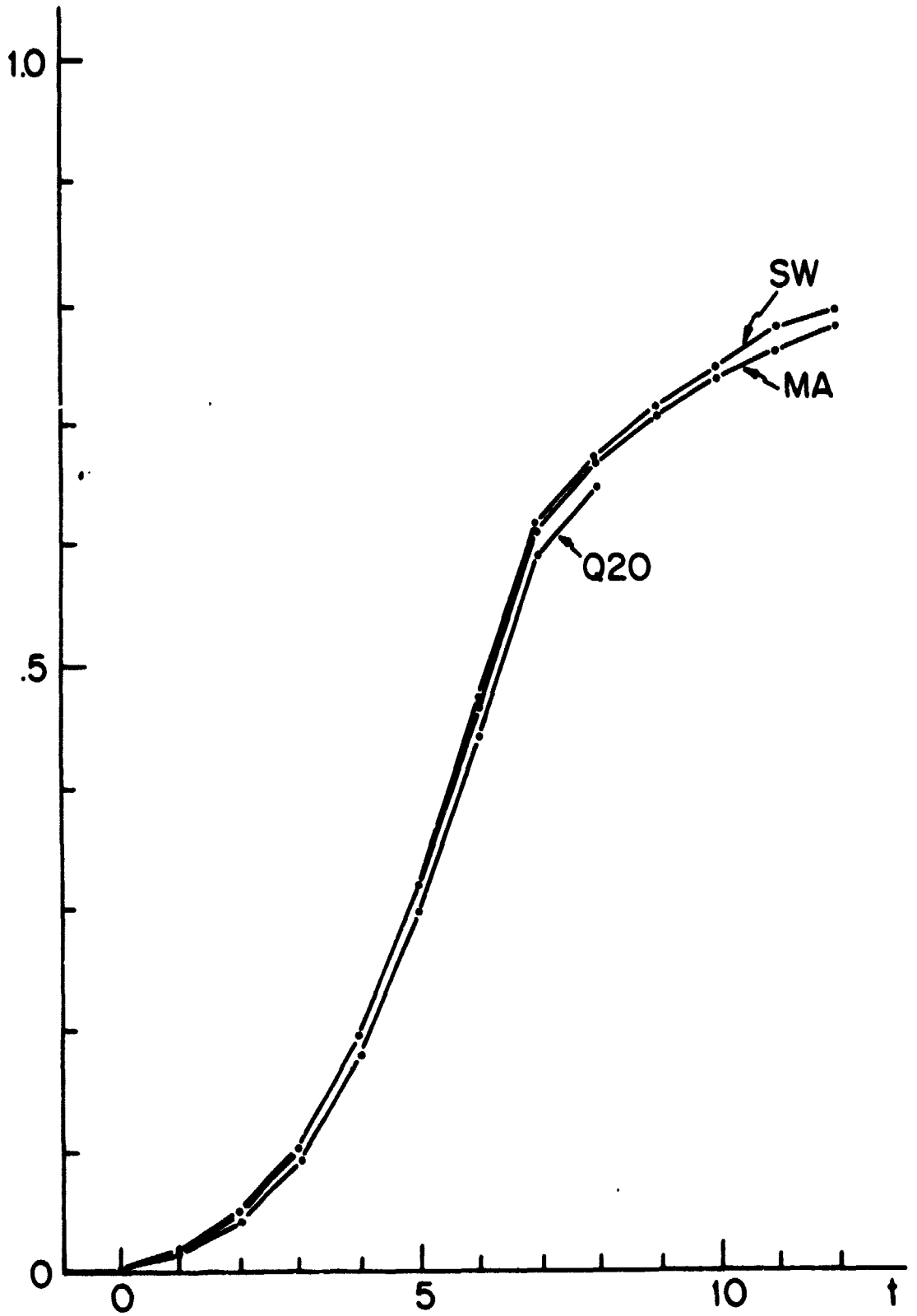


FIGURE 6-11: $b_1(t|1)$ - Sliding Window Rule and Approximation.

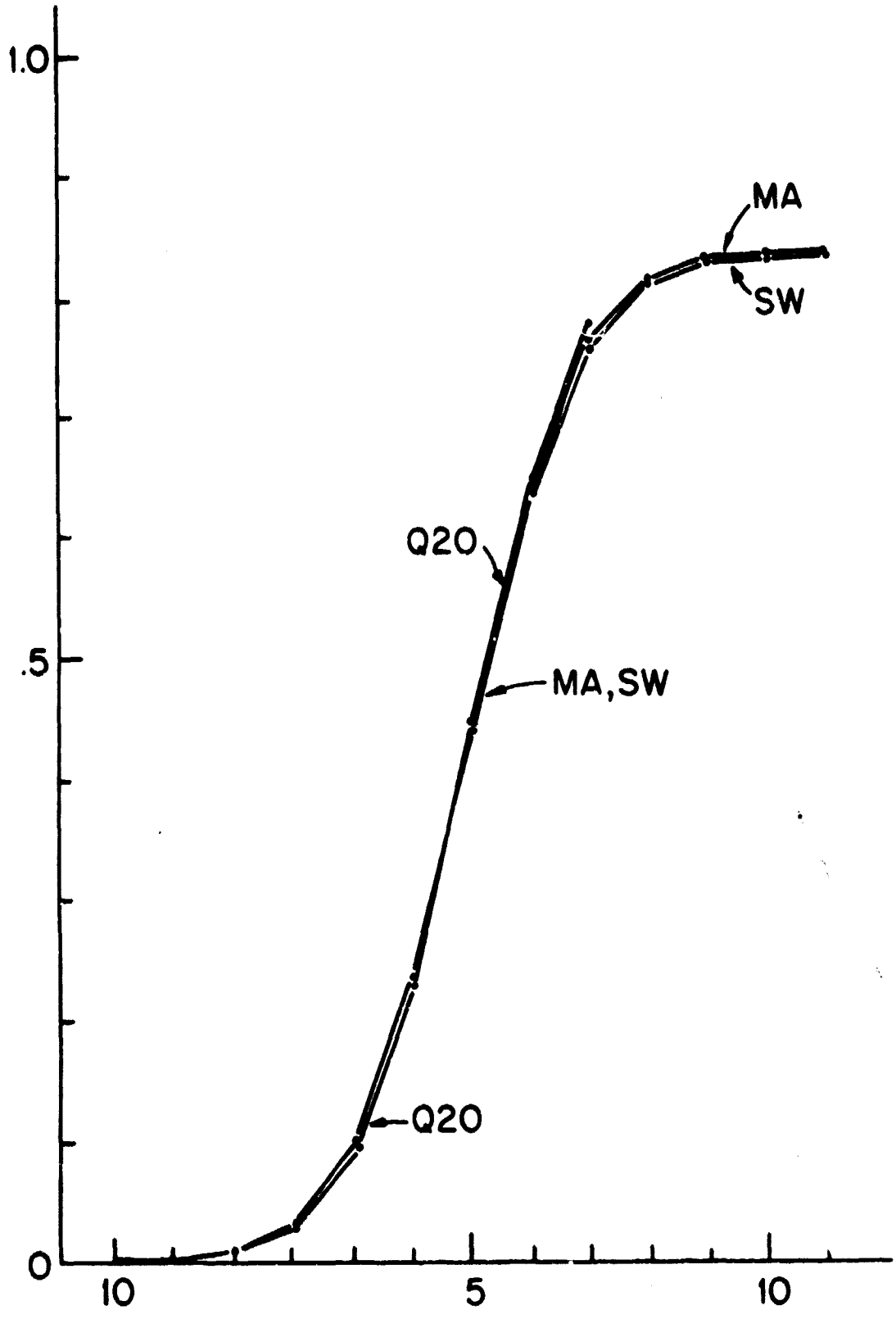


FIGURE 6-12: $b_2(t|2)$ - Sliding Window Rule and Approximation.

computed via the quadrature method using $n_L = n_H = 20$, and these are included in the above figures. (These probabilities are also marked Q20). SW may be compared with MA to determine the validity of the Markov approximation ℓ, and MA may be compared with Q20 to further assess the accuracy of the quadrature algorithm.

From the simulation results (Figure 6-8) it is evident that the Markov approximation (MA) slightly under-estimates the false alarm rate of the sliding window rule (SW). However, the response of the Markov approximation to failures is very close to that of the sliding window rule (see Figure 6-8 to 6-12). In the present example, ℓ_{W-1} is a 7-th order process, while its approximation ℓ is only of first order. In view of this fact, we can conclude that ℓ provides a very reasonable and useful approximation of ℓ_{W-1}.

The quadrature results Q20 are very close to MA. The value of P_T ranges between .998 and 1.03. This is further evidence that the quadrature algorithm is useful for obtaining estimates of probabilities. Furthermore, this indicates that the results of applying quadrature to the Markov approximation provides a good approximation of SW. Thus one overall conclusion is that the quadrature technique for calculating approximate performance using the Markov approximation to the sliding window test represents an useful and accurate method for determining the performance of failure detection rules and for comparing and optimizing such rules. We now turn to the last of these possibilities.

The SQP algorithm

The SQP algorithm is used in conjunction with the quadrature algorithm ($n_L = n_H = 20$) to find the risk-minimizing threshold for both the sliding

window rule and the rule using $\bar{z}$. The successive choices of thresholds by SQP for the two decision rules are plotted in Figures 6-13 and 6-14. The performance indices, such as the estimated mean time between false alarms (MTBFA), the detection delays ($\bar{t}_i$), and the correct detection probabilities ($P(i|i)$), along with the risks associated with these choices of thresholds for both decision rules are shown in Tables 6-5 and 6-6.

Note that we have not carried the SQP algorithm far enough so that the successive choices of thresholds are, say, within .001 of each other. In Tables 6-5 and 6-6, it is evident that towards later iterations the performance indices become relatively insensitive to small changes of the f's. This together with the fact that we are only computing an approximate Bayes risk (see Subsection 5.2.3) means that fine scale optimization is not worthwhile. Therefore, with the approximate risk, the SQP is most effeciently used to locate the zone where the minimum lies. That is, the SQP algorithm is to be terminated when it is evident that it has converged into a reasonably small region, such as in the present example (see Figure 6-13 and 6-14). Then we may choose the thresholds that give the smallest risk as the approximate solution of the minimization.

In the event that thresholds that yield the smallest risk do not provide the desired detection performance, the design parameters, L, σ, μ, and W may be adjusted and the SQP may be repeated to get a new design. A practical alternative method is to make use of the list of performance indices (Tables 6-5 and 6-6), that are the by-product of SQP, and choose a pair of

ITER	f_1	f_2	RISK	MTBFA	P(1|1)	$\bar{t}_1$	P(2|2)	$\bar{t}_2$
	8.0	11.0	8.9029	359	.868	6.29	.821	5.71
	9.5	11.5	8.9282	963	.793	8.24	.930	6.10
	9.0	12.5	8.8856	976	.917	7.75	.841	6.43
	8.5	13.5	8.9418	744	.973	7.16	.699	6.65
	9.75	13.0	8.9742	2014	.909	9.05	.888	6.74
	8.75	11.75	8.8818	676	.879	7.28	.858	6.10
1	7.563	12.654	8.9433	330	.971	6.00	.620	6.10
2	8.840	10.729	8.9091	510	.768	7.14	.914	5.72
3	8.616	12.110	8.8842	671	.912	7.17	.821	6.21
4	8.748	11.902	8.8809	703	.891	7.30	.849	6.16
5	8.801	11.978	8.8803	743	.893	7.39	.850	6.20
6	8.825	12.028	8.8802	766	.895	7.43	.850	6.22
7	9.180	12.740	8.8871	994	.875	7.93	.882	6.29
8	8.849	12.047	8.8801	783	.895	7.46	.851	6.23
	8.867	12.039						

TABLE 6-5: Performance of Sliding Window Rule with Thresholds Chosen by SQP.

ITER	f_1	f_2	RISK	MTBFA	P(1|1)	$\bar{t}_1$	P(2|2)	$\bar{t}_2$
	6.4	12.5	8.9389	761	.922	7.60	.772	6.41
	7.2	13.25	9.1026	1786	.890	9.55	.869	6.82
	6.0	11.8	8.9290	443	.914	6.74	.739	6.04
	7.0	12.0	8.9947	951	.838	8.74	.883	6.40
	5.5	11.0	8.9389	252	.907	5.83	.691	5.61
	5.7	12.2	8.9406	384	.947	6.29	.651	6.03
1	5.462	12.940	8.9643	344	.975	5.97	.541	6.09
2	5.975	10.996	8.9311	340	.869	6.56	.781	5.78
3	5.951	11.528	8.9289	395	.903	6.62	.746	5.94
4	5.776	10.771	8.9337	284	.872	6.21	.759	5.64
5	6.089	11.667	8.9279	454	.901	6.88	.763	6.03
6	6.117	11.545	8.9279	445	.891	6.90	.775	6.04
7	6.287	11.867	8.9289	563	.897	7.27	.787	6.16
	6.158	11.635						

TABLE 6-6: Performance of Decision Rule Using z with Thresholds Chosen by SQP.

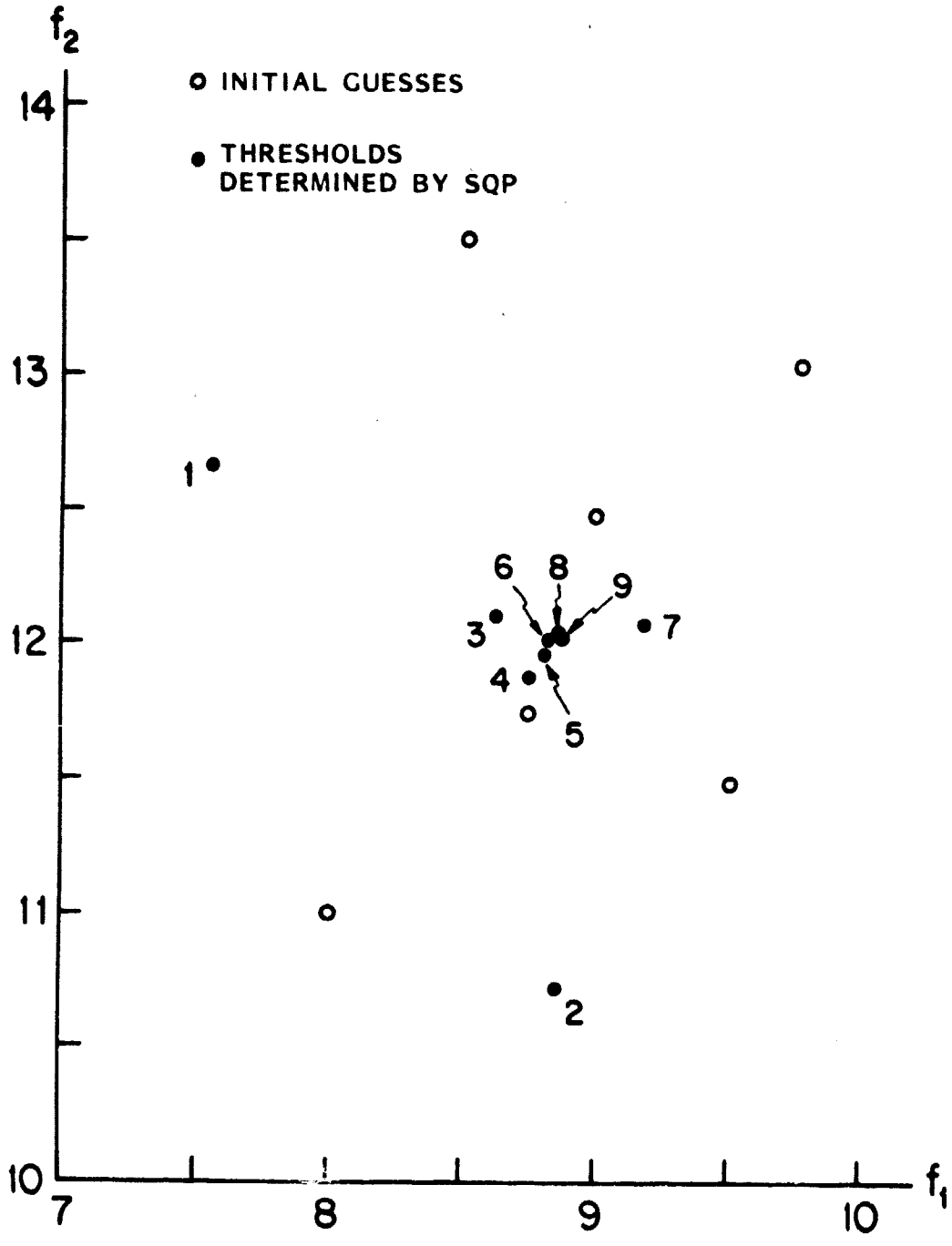


FIGURE 6-13: Thresholds of Sliding Window Rule Chosen by SQP.

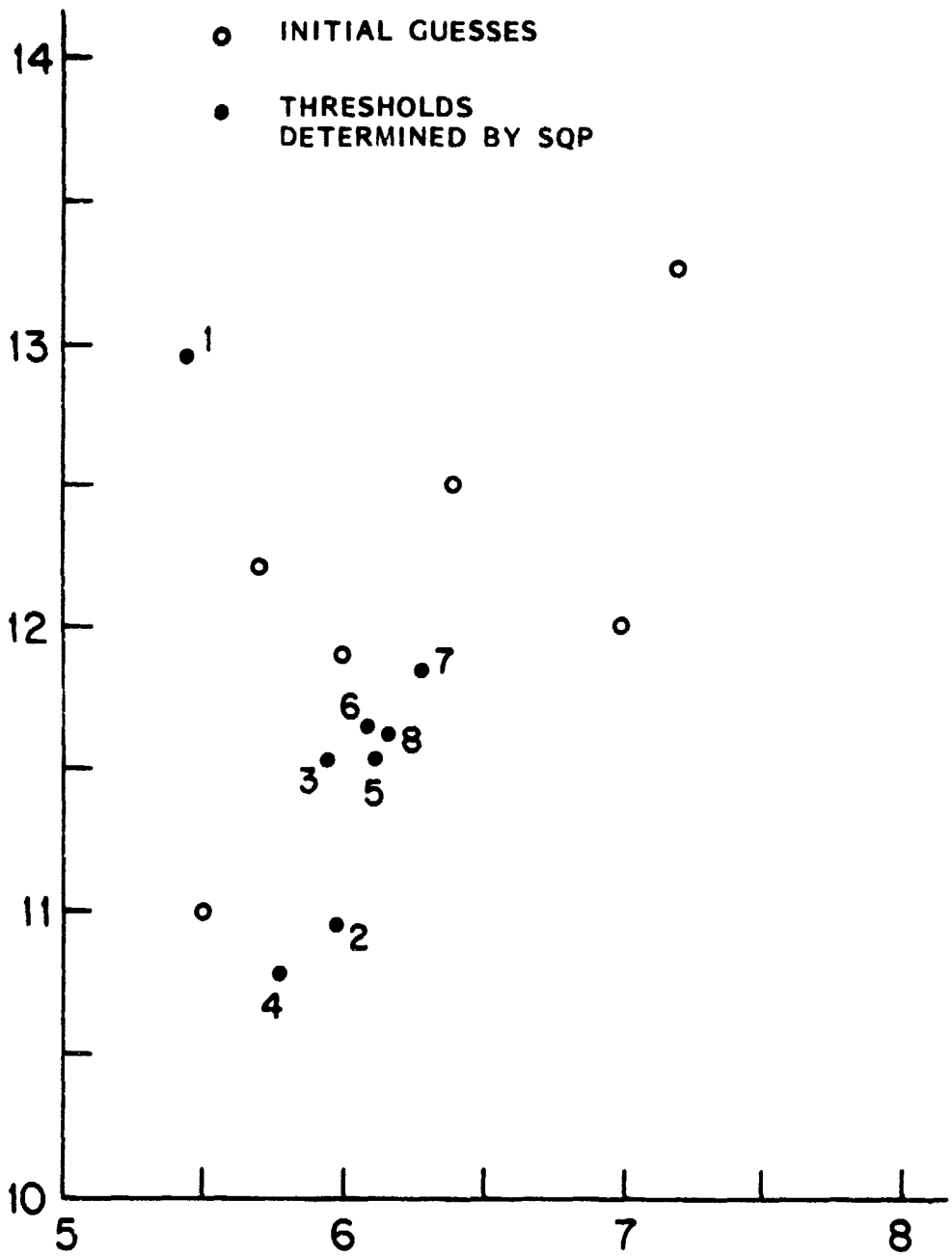


FIGURE 6-14: Thresholds Chosen by SQP for the Decision Rule Using z.

thresholds that yields the desired performance. Usually the list of performance indices provides sufficient information for deducing such a pair of thresholds heuristically. This choice of thresholds may be refined after its performance is determined via the quadrature algorithm. This approach will save the computations required to apply the SQP the second time (i.e. after we have adjusted the design parameters L, c, and μ).

Sliding window rule vs decision rule using z

Here, we will compare the performance of a sliding window rule with that of a rule using z. We will consider these rules with thresholds determined with SQP based on the same cost functions and prior probability as described in Section 6.1. The thresholds for the sliding window rule are [8.849, 12.047] (the 8-th iteration of SQP, Table 6-5), and the thresholds of the other rule are [6.287, 11.687] (the 7-th iteration of SQP, Table 6-6).

Probabilities for these two decision rules based on simulations of 10,000 trajectories are shown in Figures 6-15 to 6-19. In fact, these are the same results listed in Figures 6-3 to 6-12. Here, SW and MI are plotted on the same graph to facilitate the comparison. We note that MI has a higher false alarm rate than SW. The speed of detection for the two rules is similar. While MI has a slightly higher type-1 correct detection probability than SW, SW has a consistently higher $\beta_2(t|2)$ (type-2 correct detection probability) than MI. (Also see Tables 6-5 and 6-6).

Based on the results (Table 6-5 and 6-6) we can make the following deduction. By raising the thresholds of the rule using z appropriately, we

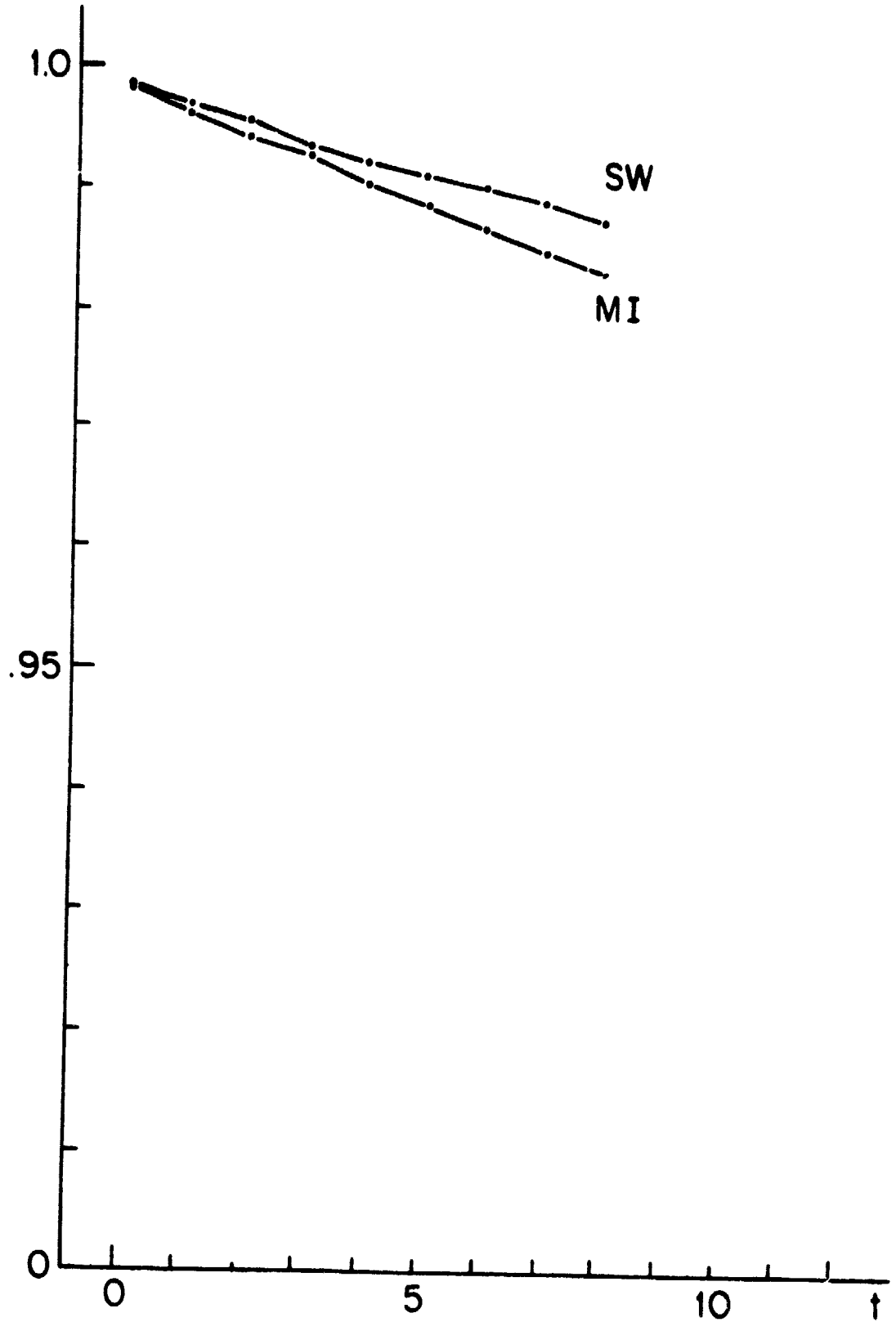


FIGURE 6-15: $b_0(t|0)$ - SW and MI.

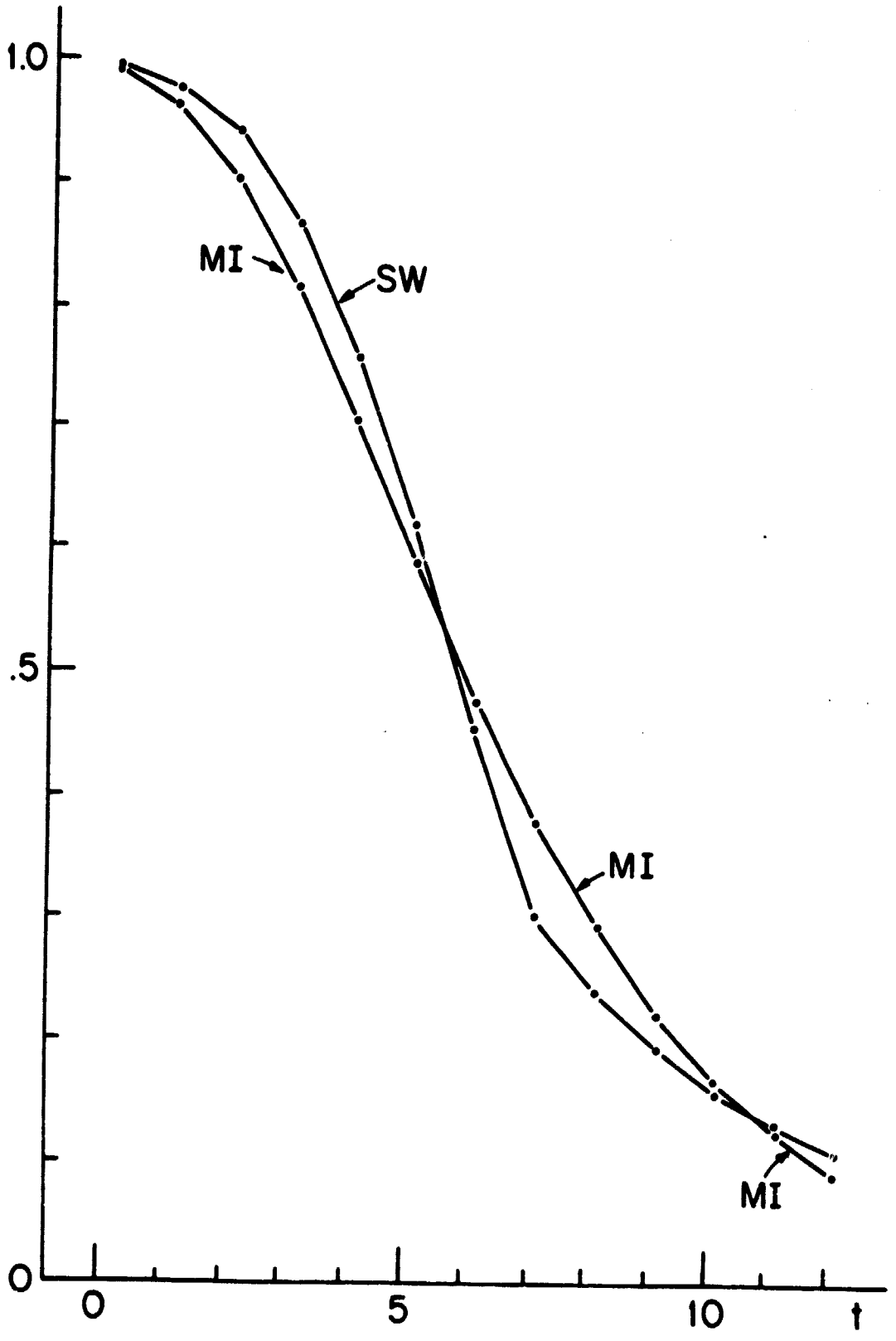


FIGURE 6-16: $b_0(t|1)$ - SW and MI.

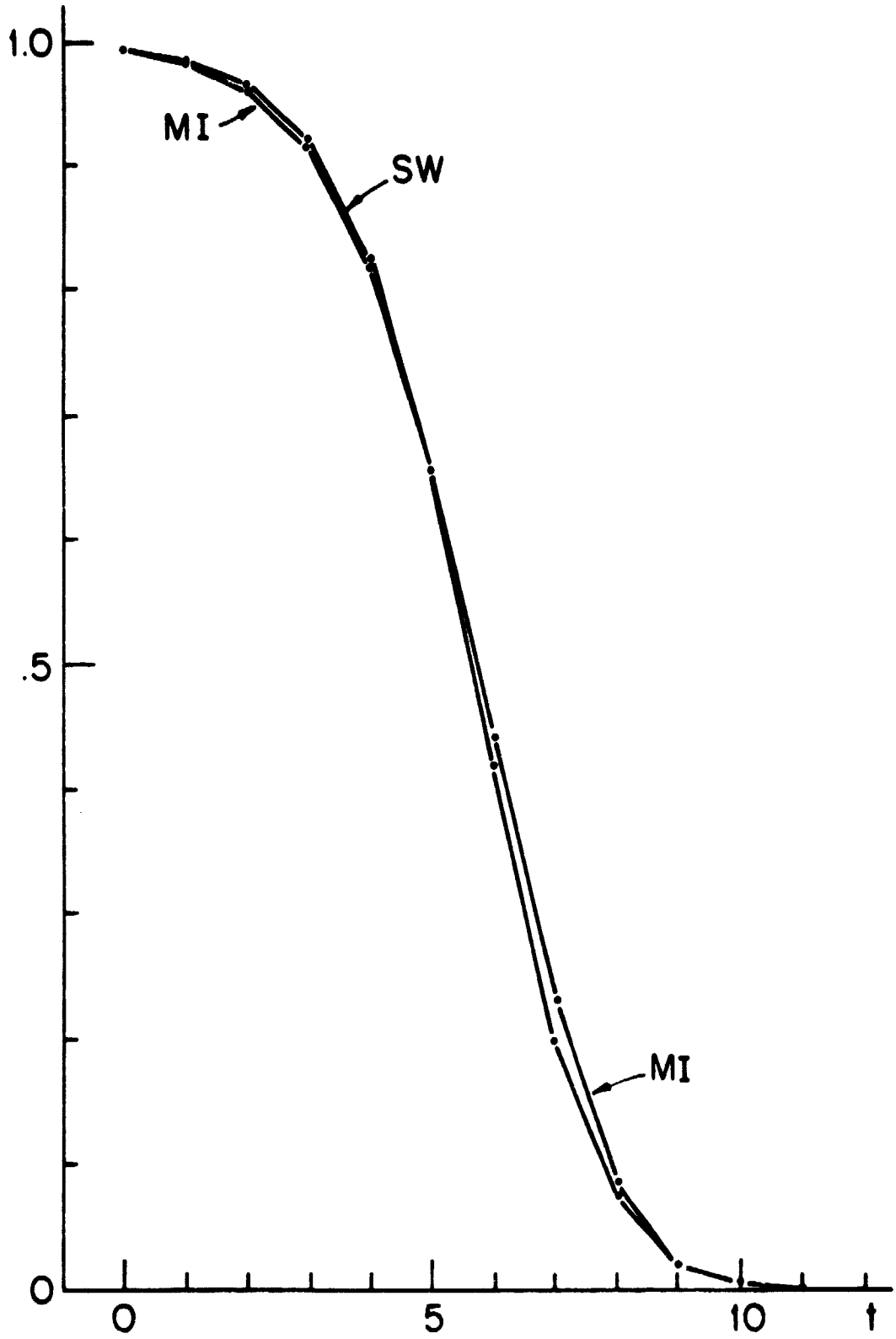


FIGURE 6-17: $b_0(t|2)$ - SW and MI.

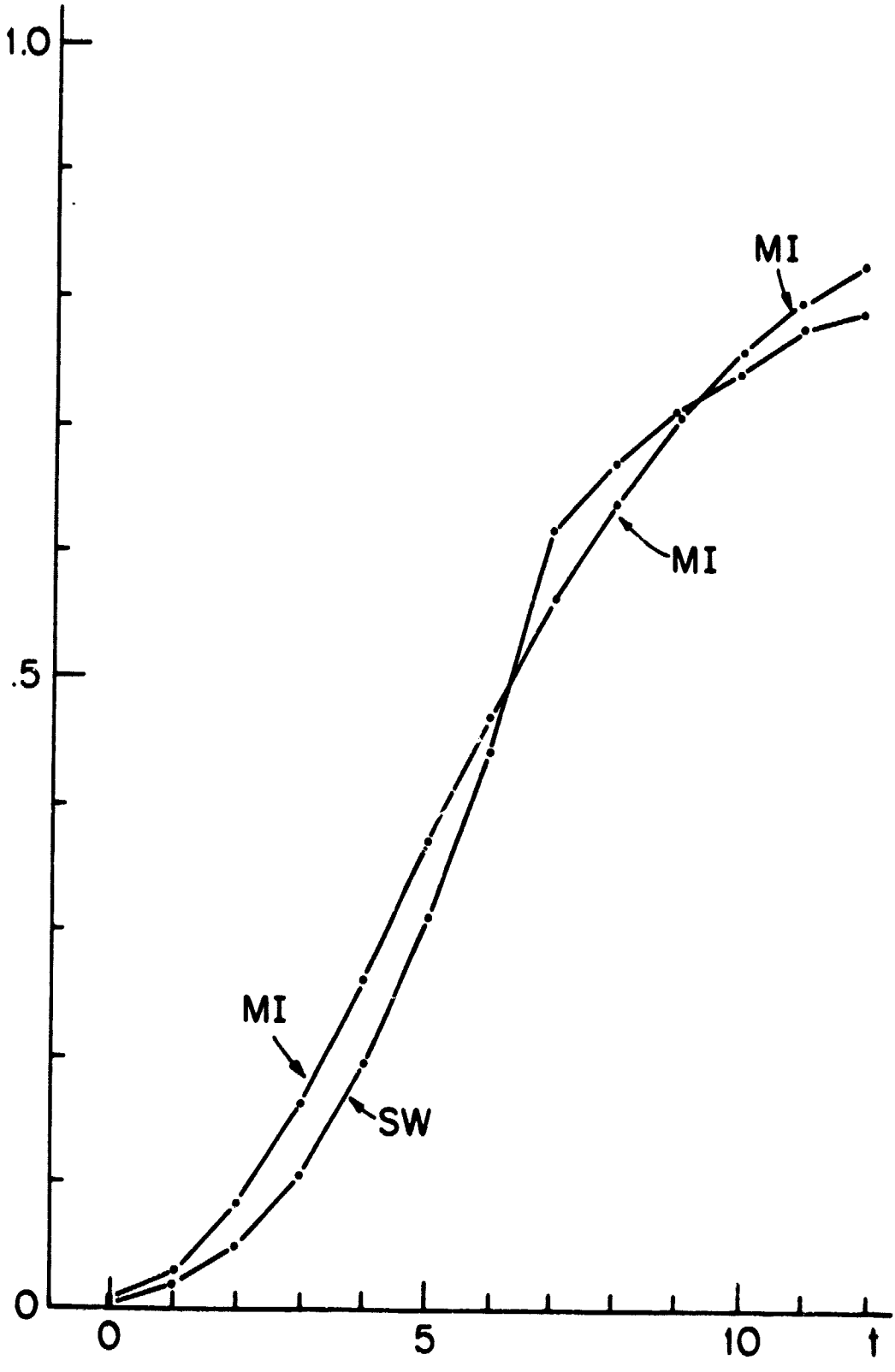


FIGURE 6-18: $\beta_1(t|1)$ - SW and MI.

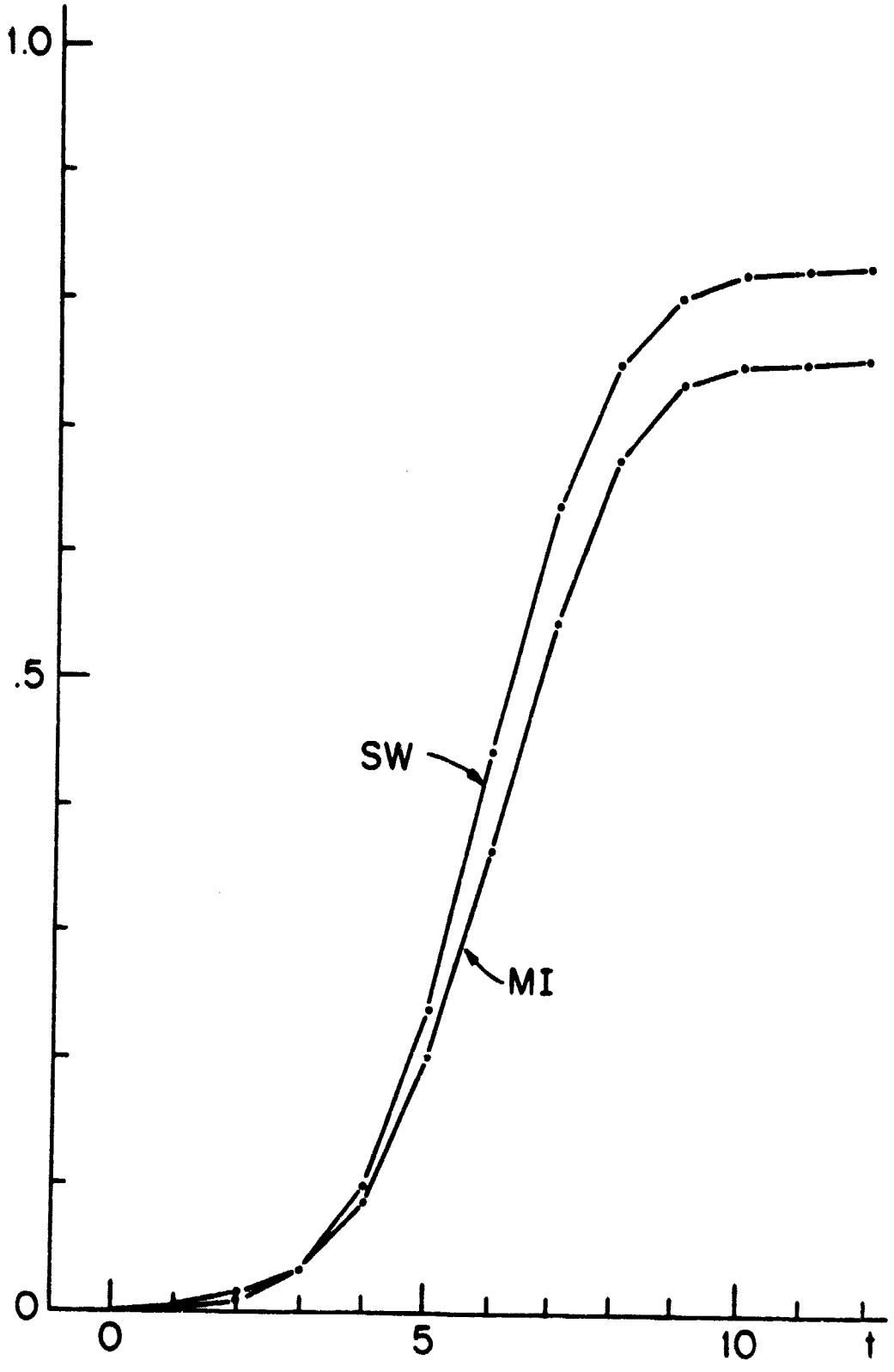


FIGURE 6-19: $\beta_2(t|2)$ - SW and MI.

can decrease the false alarm rate of MI down to that of SW with an increase in detection delay and slightly improved correct detection probability for the type-2 failure (with ramp signature). Thus, the sliding window rule is slightly superior to the rule using z in the sense that when both are designed to yield a comparable false alarm rate, the latter will have longer detection delays and slightly lower correct detection probability (for type-2 failure). In view of the fact that a decision rule using z is much simpler to implement, it is worthy of being considered as an alternative to the sliding window rule.

In summary, the result of applying our decision rule design method to the present example is very good. The quadrature algorithm has been shown to be useful, and the Markov approximation of ℓ_{W-1} by ℓ is a valid one. The SQP algorithm has demonstrated its simplicity and usefulness through the numerical example. Finally, the Markov decision statistic z has been shown to be a worthy alternative to the sliding window statistic ℓ_{W-1}.

CHAPTER 7

SUMMARY AND RECOMMENDATIONS

7.1 Summary

The goal of the research reported in this thesis is to develop a methodology for designing FDI systems that deliver good performance and are robust in the presence of modelling uncertainties. We have viewed the FDI process as consisting of two stages: a residual-generation process followed by a decision process. Since modelling errors affect residual-generation directly, the robustness issue is most effectively tackled in the design of this process. Naturally, the issue of detection performance is the main concern in designing a decision rule. Therefore, the FDI design problem is decomposed into two tasks: the design of a robust residual-generation process and the design of a high performance decision rule.

Analytical redundancy is the basis for residual-generation. In Chapter 2, we presented a general formulation of the concept of analytical redundancy for LTI systems in terms of a parity space. A redundancy relation is simply a parity relation, which has to hold in the absence of a failure and noise. When such a relation is violated, a failure is evident. The use of parity functions (or parity vectors) as residuals for FDI was also extensively discussed.

In the presence of modelling uncertainties and noise, the parity relations of the system also become uncertain. Chapter 3 was devoted to the development of an approach for determining useful parity relations for FDI. The crucial problem of determining a set of appropriate coefficients for a

parity relation was formulated as a minimax problem, the objective of which was to minimize the worst case effect of noise and modelling error on the parity relation. Therefore, residual-generation based on such parity relations is robust (or as robust as any relation can be for the particular system under consideration). The notion of signature-to-parity-error ratio was also introduced to aid in the choice of parity functions for residual-generation.

The contribution of Chapters 2 and 3 rests on the precise characterization of analytical redundancy as parity relations and the formulation of the parity coefficient design problem as a minimax optimization. These concepts have formed the basis of a new approach to the design of robust residual-generation processes. Further development of this design method is possible, and we will discuss some of the future research directions in Section 7.2.

The design of a decision rule involves resolving the tradeoff among the various detection performance issues. In this research we followed the Bayesian approach. In Chapter 4, we formulated the FDI decision process as a Bayes sequential decision problem. The cost functions and the prior probability mass function of the Bayes method could be regarded as parameters prescribing the tradeoff among the various performance issues. Although the optimal Bayes rule cannot be implemented, this formulation provides a structure from which simple suboptimal rules can be constructed.

In Chapter 5, we discussed some suboptimal decision rules that are based on the Bayes rule as well as other suboptimal rules. Just as in the

design of a Bayes rule, the design of a suboptimal rule was also formulated as a risk-minimizing problem. A quadrature algorithm was developed to compute the detection probabilities and the risks associated with low dimensional problems. Thus, with a minimization algorithm that does not require derivative information, such as the SQP, the suboptimal rule design problem may be solved numerically.

This design methodology was applied in a numerical example. The results (discussed in Chapter 6) indicate that this approach is a valid and useful one. We also note that the limitations on the dimensionality imposed by computational considerations need not lead to a corresponding severe limitation in the applicability of our technique. Specifically, our work in Chapter 2 and 3 was aimed at breaking up the dynamics of a system into low-dimensional pieces in order to isolate robust parity relations. Thus we see that using low-dimensional decision statistics serves two purposes: it allows us to address the issue of robustness and it allows us to apply our decision rule algorithm.

7.2 Future Research Directions

In the course of this study, a number of open problems have been encountered, and they were mentioned in the text of this thesis. Some directions for future research based on these problems are outlined in the following:

1) In section 3.5, we described the solution to a special case of the minimax parity coefficient design problem. A solution procedure for more general cases is needed.

2) In Section 3.3 we indicated that if we postulate a PDF over Γ, the parity coefficient design problem can be re-formulated as a minimization problem, which is much simpler to solve. This approach should be examined in the future.

3) Because a parity relation with a large signature-to-parity-error ratio (π) is desirable for FDI, we may re-define the objective of the parity coefficient design problem so that we consider

$$\max_{\alpha,\beta} \min_{\gamma\in\Gamma} \pi(\alpha,\beta,\gamma,x_0,\tilde{u}_0,i)$$

$$\text{s.t.} \quad \alpha\alpha'=1$$

where i denotes the failure of interest. This problem is generally more difficult than the minimax problem, because the objective function π is more complex.

4) The parity coefficient design procedure examined in Chapter 3 yields parity relations that are most suitable for robust open-loop residual-generation. The problem of determining parity coefficients (relations) for robust closed-loop residual-generation should be addressed.

5) In the present study, we did not consider in detail how to choose a set of "best" parity functions as measured by π or e^* (the parity error) for the FDI of a given set of failures. A systematic method for selecting this set of parity relations is a useful tool to be developed in the future.

6) The detection performance indices (such as correct detection probabilities), associated with the decision rules of Chapter 5 are based

on exact characterization of failure signatures. The effect of modelling error is not accounted for. It will be useful to make use of information such as π to couple the effect of modelling uncertainties into the detection performance indices. Then we can consider designing decision rules that minimize the risks based on the modified performance indices.

7) The quadrature algorithm developed in Chapter 5 provides reasonable estimates of the probabilities. However, it consumes too much computation time. An improvement of this algorithm aimed at reducing computations is desirable. For example, we may consider a better placement of the grid points of the quadrature so that fewer points will be needed. With reduced computational requirement, the quadrature algorithm may be used for higher dimensional problems. In addition, the utilization of other 1-dimensional quadrature formulas in place of the Laguerre and Hermite formulas used in the present quadrature algorithm should be explored in an effort to arrive at a more efficient integration formula that is applicable to higher-dimensional problems as well as the 2-dimensional case considered here.

8) In Chapter 6, the (implementable) Markov decision statistic z was shown to be simple and useful. In order to generate such a statistic, a choice of the matrices $\tilde{A}$ and $\tilde{B}$ is needed. A procedure for selecting these matrices for high detection performance is needed.

9) More experimentation is needed to confirm the general conclusions of our study of the decision rule optimization algorithm.

APPENDIX A: Solution Procedure for the Minimax Problem (3-51)

Consider the optimization problem

$$\min_{x \in X} \max [f_1(x), f_2(x)] \tag{A-1}$$

where $X \subseteq R^n$. Both f_1 and f_2 are continous over X which is a connected subset of R^n. For each f_i, there is a subset T_i of X such that for any point $x_0 \in T_i$ and any point $x \in X$, $f_i(x_0) \leq f_i(x)$. That is, T_i contains the global minima of f_i over X. We will assume that f_i has no other local minima.

We can show that the solution to (A-1) can be determined as follows:

1) By defining

$$h(x) = \max[f_i(x), f_2(x)]$$

we can re-write (A-1) as

$$\min_{x \in X} h(x) \tag{A-2}$$

Let Ω be the set

$$\Omega = \{x: x \in T_i \text{ and } f_i(x) \geq f_{3-i}(x), \quad i=1,2\}$$

It is clear that when Ω is not empty, it contains the minimum of h over X. Assuming Ω is not empty, the solution of (A-2) is simply the element $x^* \in \Omega$ such that

$$h(x^*) = \min_{x \in \Omega} h(x) \tag{A-3}$$

Note that X^* may not be unique. The solution of (A-3) is easy to compute if each f_i has a single global minimum, because Ω will contain at most two points and the search required in (A-3) is extremely simple. When Ω is empty, the minimax solution can be found using the next step.

2) Define

$$\Lambda = \{x: x \in X \text{ and } f_1(x) = f_2(x)\} \tag{A-4}$$

and let Λ^c denote the complement of Λ in X. Consider an element $\bar{x} \in \Lambda^c$. Because f_1 and f_2 are continuous and X is connected, we can find a neighborhood N around $\bar{x}$ in Λ^c such that either $f_1(x)>f_2(x)$ or $f_2(x)>f_1(x)$ for all $x \in N$. Without loss of generality, we can assume $f_1(x)>f_2(x)$ in N. If a solution is not found by using step 1, we only need to consider $\bar{x} \notin T_i$. In this case, there is some other point x^0 in N (hence in Λ^c) such that $f_1(x^0)<f_1(\bar{x})$ (since f_1 has no local minima). Therefore, $\bar{x}$ cannot be a solution of (A-2) and Λ^c does not contain the solution. The solution must lie in Λ. In this case, (A-1) becomes

$$\begin{aligned} &\min \quad f_1(x) \\ &\text{s.t} \quad x \in X \\ &\qquad f_1(x)-f_2(x)=0 \end{aligned} \tag{A-5}$$

and we have a constrained minimization problem. Note that the objective function of (A-5) may be replaced by $f_2(x)$.

Now we will apply the above result to the minimax problems considered in Section 3.5. First, consider (3-51), where f_1, f_2 and X are

$$f_i(\alpha') = \alpha S(\gamma^i)\alpha' , \qquad i=1,2,$$

$$X = \{\alpha' : \alpha\alpha'=1\}$$

where $S(\gamma^i)$ is an nxn symmetric positive definite matrix and α is a (row) n-vector. It is well-known that $S(\gamma^i)$ has a complete set of orthonormal eigenvectors. The minimum value of f_i is clearly the smallest eigenvalue of $S(\gamma^i)$. Now we will show that f_i has no local minimum other than the global ones. It is clear that the eigenvectors $y_1,\ldots,y_n$ of $S(\gamma^i)$ represent all possible local minima of f_i over X. Let y_1 correspond to the eigenvalue σ_1 of $S(\gamma^i)$ which is greater than the smallest eigenvalue σ_{min} (which associated eigenvector y_{min}). By taking $\alpha' = ay_1 + by_{min}$ with $b\neq 0$ and $a^2+b^2=1$, we have $f_i(\alpha') = a^2\sigma_1 + b^2\sigma_{min} < \sigma_1$. Thus, y_1 cannot be a local minimum and f_1 has no local minimum other than y_{min}. Consequently, the two-step solution technique is applicable to (3-51).

When the smallest eigenvalue of $S(\gamma^i)$ is not repeated, f_i has one global minimum at y_{min}. (Due to symmetry, we can consider y_{min} only and not $-y_{min}$.) If this is true for f_1 and f_2, Ω has at most two points and the solution of (A-3) (i.e. in step 1) can be readily determined. When σ_{min} of $S(\gamma^1)$ is repeated, we can also show that we only need to consider at most two points

of Ω in step 1. Suppose σ_{min} is of multiplicity m, and we let Y be the matrix of the m eigenvectors associated with σ_{min}. (Note that Y is not unique, but it has rank m.) Then

$$T_1 = \{\alpha' : \alpha' = Yz, \text{ where } z \in R^n \text{ and } z'z=1\}$$

To determine if we can find an $\alpha' \in T_1$ such that $f_1(\alpha') \geq f_2(\alpha')$, it is only necessary to check if $f_1(\bar{\alpha}') \geq f_2(\bar{\alpha}')$ where $\bar{\alpha}'$ is the solution of

$$\begin{aligned} \min_{\alpha} \quad & \alpha S(\gamma^2)\alpha \\ \text{s.t.} \quad & \alpha' = Yz \\ & z'z=1 \end{aligned}$$

Equivalently, $\bar{\alpha}' = Y\bar{z}$, where $\bar{z}$ solves

$$\min_{z'z=1} z'[Y'S(\gamma^2)Y]z \qquad \text{(A-6)}$$

The solution to (A-6) is, of course, the eigenvector of $Y'S(\gamma^2)Y$ (which is mxm, symmetric, positive-definite) associated with the smallest eigenvalue. Note that $\bar{z}$ may not be unique (due to repeated eigenvalues of $Y'S(\gamma^2)Y$), and hence, $\bar{\alpha}'$ may not be unique. Since all such $\bar{\alpha}'$ are equivalent (give the same value $f(\bar{\alpha}')$), we only have to consider one of them. As a result, there will be at most two points in Ω (which may be empty) in step 1.

Next, we will apply the above solution procedure to the case where the parity structure contains actuator inputs. The minimax problem is of the form

$$\min_{\alpha,\lambda} \max_{\gamma \in [\gamma^1, \gamma^2]} [\alpha\ \lambda]\ S(\gamma)\ [\alpha\ \lambda]' \qquad \text{(A-7)}$$

$$\text{s.t.} \quad \alpha\alpha' = 1$$

where S is the symmetric positive definite matrix given by (3-30) and λ is a scalar. Assuming S is quadratic in γ (a scalar parameter), (A-7) is equivalent to

$$\min_{\alpha,\lambda} \max [\alpha\lambda] S(\gamma^1) [\alpha\lambda]',\ [\alpha\lambda] S(\gamma^2) [\alpha\lambda]' \qquad \text{(A-8)}$$

$$\text{s.t.} \quad \alpha\alpha' = 1$$

with $X = \{[\alpha\lambda]' : \alpha\alpha' = 1\}$. Therefore, the 2-step solution procedure described above applies if we can show that $[\alpha\lambda] S(\gamma^i) [\alpha\lambda]'$ has no local minima other than the global one over X. According to multiplier theory [14], the necessary condition for a local minimum of $[\alpha\lambda] S(\gamma^i) [\alpha\lambda]'$ to exist over X is

$$[\alpha\ \lambda] \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \theta [\alpha\ 0] \qquad \text{(A-9)}$$

where we have shown S in the partitional form, and θ is a non-zero scalar. Since S is positive-definite, (A-9) can be re-stated as

$$\alpha [S_{11} - S_{12}\, S_{22}^{-1}\, S_{21}] = \theta\ \alpha \qquad \text{(A-10)}$$

$$\lambda = -S_{22}^{-1}\, \alpha\, S_{12} \qquad \text{(A-11)}$$

Therefore, in studying the minima of $[\alpha\lambda]S(\gamma^i)[\alpha\lambda]'$ we only need to consider λ of the form (A-11). Using (A-11), we have

$$[\alpha\lambda]S(\gamma^i)[\alpha\lambda]' = \alpha[S_{11}-S_{12}S_{22}^{-1}S_{21}]\alpha$$

Then, we can readily deduce that the global minimum is given by $[\bar{\alpha}, \lambda(\bar{\alpha})]$, where $\bar{\alpha}$ is the eigenvector of $[S_{11}-S_{12}S_{22}^{-1}S_{21}]$ (symmetric, positive-definite) corresponding to the minimum eigenvalue and $\lambda(\bar{\alpha})$ is given by (A-11). Moreover, $[\alpha\lambda]S(\gamma^i)[\alpha\lambda]'$ has no local minima, because $\alpha[S_{11}-S_{12}S_{22}^{-1}S_{21}]\alpha$ has no local minima.

APPENDIX B: The Convergence of $\tilde{J}_k(0)$ to $\tilde{J}_\infty(0)$

Let $\Phi^* = (\phi^*(1;r(1)),\ldots,\phi^*(k;r(1),\ldots,r(k)),\ldots)$ be the optimal stopping rule for the non-truncated sequential problem. Define $\Phi^{*,K} = (\phi^{*,K}(1,r(1)),\ldots,\phi^{*,k}(k;r(1),\ldots,r(k)),\ldots)$ such that

$$\phi^{*,K}(k;r(1),\ldots,r(k)) = \phi^*(k;r(1),\ldots,r(k)), \quad k=1,\ldots,k-1$$

$$\phi^{*,K}(K,r(1),\ldots,r(k)) = 1$$

That is, $\Phi^{*,K}$ is the same as Φ^*, except the former imposes mandatory stopping at time K, and $\phi^{*,K}(k;r(1),\ldots,r(k))$, $k>K$ is arbitrary. Since (Φ^*,D^*) is optimal for the non-truncated problem, we have the difference Δ:

$$\Delta = U_s(\Phi^{*,K},D^*) - U_s(\Phi^*,D^*) \geq 0 \tag{B-1}$$

Furthermore, from (4-16), we have

$$\begin{aligned}|\Delta| \leq & \sum_{i=0}^{M}\sum_{\tau=1}^{k}\mu'(k;i,\tau)\sum_{k=K}^{\infty}E_{i,\tau}\psi^*(k;r(1),\ldots,r(k))[L((i,\tau),d^*(K;r(1),\ldots,r(K))) \\ & \qquad + c(i)(k-\tau)] \\ & + \sum_{i=0}^{M}\sum_{\tau=1}^{\infty}\mu'(\infty,i,\tau)\sum_{k=K}^{\infty}E_{i,\tau}\psi^*(k;r(1),\ldots,r(k))[L((i,\tau),d^*(k;r(1),\ldots,r(k)))] \\ & \qquad + \sum_{\substack{k=\\ \max[\tau,k]}}^{\infty}E_{i,\tau}\psi^*(k;r(1),\ldots,r(k))c(i)(k-\tau)\end{aligned} \tag{B-2}$$

Note that in order for a sequential risk to be finite (as is true with $U_s(\Phi^*,D^*)$), $\sum_{k=K}^{\infty}E_{i,\tau}\psi(k;r(1),\ldots,r(k)) \to 0$ as $K\to\infty$ and

$\sum_{k=1}^{\infty} k\, E_{i,\tau}\psi(k;r(1),\ldots,r(k))$ must be finite. (Note that in (B-2), $c(0)=0$ because there is no delay cost if no failure has occurred.) Since L and $c(i)$ are bounded, the two terms in (B-2) that are due to L and the term due to $c(i)(K-\tau)$ vanish as $K\to\infty$. Now consider the remaining term in (B-2):

$$\left| \sum_{i=0}^{M} \sum_{\tau=0}^{\infty} \mu'(\infty;i,\tau) \sum_{k=\max[\tau,k]}^{\infty} E_{i,\tau}\psi^*(k;r(1),\ldots,r(k))c(i)(k-\tau) \right|$$

$$\leq \sum_{i=0}^{M} \sum_{\tau=0}^{\infty} \mu'(\infty;i,\tau)c(i)\tau \sum_{k=K}^{\infty} E_{i,\tau}\psi^*(k;r(1),\ldots,r(k))$$

$$+ \sum_{i=0}^{M} \sum_{\tau=1}^{\infty} \mu'(\infty;i,\tau)c(i) \sum_{k=K}^{\infty} k\, E_{i,\tau}\psi^*(k;r(1),\ldots,r(k)) \qquad \text{(B-3)}$$

From the prior discussion both terms on the right hand side of (A-3) vanish as $k\to\infty$. Therefore $\Delta\to 0$ as $k\to\infty$. The fact that $\phi^{*,K}$ belongs to the class of stopping rules that terminate sampling at or before K implies $U_s(\phi^{*,K},D^*) \geq \tilde{J}_k(0)$. Using (4-44) and (B-1) we can deduce

$$\tilde{J}_\infty(0) \leq \tilde{J}_K(0) \leq U_s(\phi^{*,K},D^*) = \tilde{J}_\infty(0) + \Delta$$

As $k\to\infty$, $\Delta\to 0$ so that $\tilde{J}_K(0) \to \tilde{J}_\infty(0)$.

REFERENCES

[1] A.H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York. (1970)

[2] F.A. Evans and J.C. Wilcox, "Experimental Strapdown Redundant Sensor Inertial Navigation System," J. of Spacecraft and Rockets, Vol. 7, No. 9, pp. 1070-1074, Sept. (1970).

[3] J.P. Gilmore and R.A. McKern, "A Redundant Strapdown Inertial Reference Unit (SIRU)," J. of Spacecraft and Rockets, Vol. 9, No. 1, pp. 39-47, Jan., (1972).

[4] A.S. Willsky and H.L. Jones, "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems," IEEE Trans. Aut. Control, AC-21, 108-112, Feb. (1976).

[5] A.S. Willsky, E.Y. Chow, S.B. Gershwin, C.S. Greene, P.K. Houpt, and A.L. Kurkjian, "Dynamic Model-Based Techniques for the Detection of Incidents on Freeways," IEEE Trans. Auto. Control, Vol. AC-25, No. 3, pp. 347-360, Jun. (1980).

[6] D.E. Gustafson, A.S. Willsky and J.Y. Wang, Final Report: Cardiac Arrhythmia Detection and Classification Through Signal Analysis, The Charles Stark Draper Lab., Cambridge, MA, Report No. R-920, July (1975).

[7] R.V. Beard, Failure Accommodation in Linear Systems Through Self-Reorganization, Man Vehicle Lab., M.I.T., Cambridge, MA, Report MVT-71-1, Feb. (1971).

[8] H.L. Jones, Failure Detection in Linear Systems, Ph.D. Thesis, Dept. of Aeronautics and Astronautics, M.I.T., Cambridge, MA., Sept. (1973).

[9] A.S. Willsky, "A Survey of Design Methods for Failure Detection in Dynamic Systems," Automatica, Vol. 12, pp. 601-611, (1976).

[10] A. Wald, Sequential Analysis, Wiley, New York, (1947).

[11] J.C. Deckert, M.N. Desai, J.J. Deyst and A.S. Willsky, "F-8 DFBW Sensor Failure Identification Using Analytic Redundancy," IEEE Trans. Auto. Control, Vol. AC-22, No. 5, pp. 795-803, Oct. (1977).

[12] J.E. Potter and J.C. Deckert, "Minimax Failure Detection and Identification in Redundant Gyro and Accelerometer Systems," J. of Spacecraft and Rockets, Vol. 10, No. 4, pp. 236-243, April (1973).

[13] J.E. Potter and M.C. Suman, "Thresholdless Redundancy Management with Arrays of Skewed Instruments," AGARDOGRAPH-224, Integrity in Electronic Flight Control Systems, pp. 15-1 - 15-25, (1977).

[14] D.P. Bertsekas, Notes on Nonlinear Programming and Discrete-Time Optimal Control, M.I.T. Lab. for Inform. and Dec. Sys., Report 919, Cambridge, MA, July (1979).

[15] R.W. Brockett, Finite Dimensional Linear Systems, Wiley, New York, (1970).

[16] K.C. Daly, E. Gai and J.V. Harrison, "Generalized Likelihood Test for FDI in Redundant Sensor Configurations," J. Guidance and Control, Vol. 2, No. 1, pp. 9-17, Jan. - Feb. (1979).

[17] D.P. Bertsekas, Dynamic Programming and Stochastic Control, Academic Press, New York, (1976).

[18] A. Wald and J. Wolfowitz, "Optimum Character of the Sequential Probability Ratio Test," Annals Maths. Stat., Vol. 19, pp. 581-586. (1948).

[19] A.H. El-Sawy and V.D. Vandelinde, "Robust Sequential Detection of Signals in Noise," IEEE Trans. Information Theory, Vol. IT-25, No. 3, pp. 346-353, May (1979).

[20] T.W. Anderson, "A Modification of the Sequential Probability Ratio Test to Reduce the Sample Size," Annals. Math. Stat., Vol. 31, pp. 165-197, (1960).

[21] T.T. Chien and M.B. Adams, "A Sequential Failure Detection Technique and Its Applications," IEEE Trans. Aut. Control, Vol. AC-21, pp. 750-757, Oct. (1976).

[22] D. Blackwell and M.A. Girshick, Theory of Games and Statistical Decisions, Wiley, New York (1951).

[23] R.A. Roberts and C.T. Mullis, "A Bayes Sequential Test of M Hypothesis," IEEE Trans. Information Theory, pp. 91-94, Jan. (1970).

[24] G. Schwarz, "Asymptotic Shapes of Bayes Sequential Testing Regions," Annals Maths. Stat., Vol. 33, pp. 225-265, (1962).

[25] A.N. Shiryaev, "On the Detection of Disorder in a Manufacturing Process, I.," Theory of Prob. and Appl. Vol. 8, No. 3, pp. 247-265, (1963).

[26] A.N. Shiryaev, "On the Detection of Disorder in a Manufacturing Process, II," Theory of Prob. and Appl., Vol. 8, No. 4, pp. 402-413, (1963).

[27] I.B. MacNeill, "Test for Change of Parameter at Unknown Times and Distributions of Some Related Functions on Brownian Motion," Annals Maths. Stat., Vol. 2, No. 5, pp. 950-962, (1974).

[28] H. Cheroff and S. Zacks, "Estimating the Current Mean of a Normal Distribution Which is Subjected to Changes in Time," Annals Maths. Stat., Vol. 35, No. 3, pp. 999-1018, Sept. (1964).

[29] A.N. Shiryaev, "On Optimum Methods in Quickest Detection Problems," Theory of Prob. and Appl., Vol. 8, No. 1, pp. 22-46, (1963).

[30] I. Shimi, "The Bayesian and Nonparametric Approach to Reliability Studies: A Survey of Recent Work," Conference on the Theory and Applications of Reliability with Emphasis on Bayesian and Nonparametric Methods, Academic Press, pp. 5-47, (1977).

[31] T.S. Ferguson, Mathematical Statistics, Academic Press, New York, (1967).

[32] H.L. Van Trees, Detection, Estimation and Modulation Theory, Vol. I, Wiley, New York (1968).

[33] R. Bellman, Dynamic Programming, Princeton Univ. Press, (1956).

[34] I.F. Blake and W.C. Lindsey, "Level-Crossing Problems for Random Processes," IEEE Trans. Information Theory, Vol. IT-19, No. 3, pp. 295-315, May (1973).

[35] R.G. Gallager and C.W. Helstrom, "A Bound on the Probability that a Gaussian Process Exceeds a Given Function," IEEE Trans. Information Theory, Vol. IT-15, No. 1, pp. 163-166, Jan. (1969).

[36] D.H. Bhati, "Approximations to the Distribution of Sample Size for Sequential Tests, I. Tests of Simple Hypotheses," Biometrika, Vol. 46, pp. 130-138, (1973).

[37] B.K. Ghosh, "Moments of the Distribution of Sample Size in a SPRT," American Statistical Association Journal, Vol. 64, pp. 1560-1574, (1969).

[38] B.K. Walker, A Semi-Markov Approach to Quantifying Fault-Tolerant System Performance, Sc.D. Thesis, Dept. of Aero. and Astro., M.I.T., Jul. (1980).

[39] P.J. Davis and P. Rabinowitz, Numerical Integration, Blaisdell Publishing Company, Waltham, Mass., (1967).

[40] A.H. Stroud and D. Secrest, Gaussian Quadrature Formulas, Prentice Hall, Englewood Cliffs, N.J., (1966).

[41] S. Haber, "Numerical Evaluation of Multiple Integrals," SIAM Review, Vol. 12, No. 4, pp. 481-526, Oct. (1970).

[42] D.H. Winfield, Function Minimization Without Derivatives by a Sequence of Quadratic Programming Problems, Harvard Univ., Division of Engineering and Applied Physics, Technical Report No. 537, Cambridge, Mass., Aug. (1967).