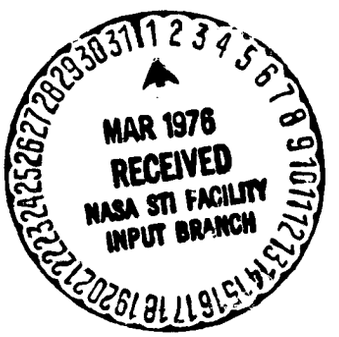
ELECTRONIC SYSTEMS LABORATORY
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Research Status Report #2

A DUAL-MODE GENERALIZED LIKELIHOOD RATIO APPROACH TO SELF-REORGANIZING DIGITAL FLIGHT CONTROL SYSTEM DESIGN

NASA Langley Grant NSG-1112

Covers work from March 15, 1975 to November 30, 1975



Prepared by:

Ramon Bueno
Edward Chow
Dr. Stanley B. Gershwin
Prof. Alan S. Willsky (Principal Investigator)

Submitted to:

(1) NASA Scientific and Technical Information Facility
Post Office Box 33
College Park, Maryland 20740

(2) Dr. Raymond C. Montgomery
Theoretical Mechanics Branch
Flight Dynamics and Control Division
NASA Langley Research Center
Hampton, Virginia 22365

I. Introduction

In this report we summarize the results of the research efforts under Grant NSG-1112 from March 15, 1975 through November 30, 1975. The personnel involved in this project during this period were Prof. A.S. Willsky, Dr. S. B. Gershwin, Mr. R. Bueno, and Mr. E. Chow. Mr. Bueno and Mr. Chow are graduate students working towards S.M. degrees and theses based on their work on this project will be forthcoming (February 1965 for Chow and September 1976 for Bueno).

Before we outline the report, we first describe several other activities related to this grant. As we see it, the purpose of this research effort is to perform a fundamental study of the problem of failure detection and reliable system design for digital aircraft control systems. The research efforts described herein represents a major step in this study; and at the end of this report we will outline several of the steps which will be examined next. In addition to this work, Prof. Willsky undertook a detailed survey of failure detection methods, and this effort culminated in the survey paper [3], which is included as Appendix A. Also, during this time period, close contact was established with staff at the Charles Stark Draper Laboratory (in particular Mr. J. C. Deckert, Dr. J. J. Deyst, Jr., and Dr. M. Desai) working on NASA Langley Contract NAS1-13914. Specifically, Prof. Willsky has been involved on a regular basis as a consultant, and Mr. E. Chow will join the CSDL research staff on this project on a full-time basis beginning in mid-January. This project, which is of a more applied nature than Grant NSG-1112, has complemented the research at the Electronic Systems Laboratory quite well. The CSDL program has provided a test-bed for

many of the concepts developed at ESL; it has also suggested several new, fundamental issues which will be explored at ESL; the CSDL project has provided a learning experience for graduate students who can then fit into the ongoing CSDL project; and, finally, the results of the CSDL study will provide several important pieces in our overall effort to develop a fault-tolerant control system design methodology. We feel that the present cooperative arrangement between ESL and CSDL provides an ideal balance for research and development in this area.

As background for the work described in this report, we refer the reader to the earlier research report [2]. We review some of the notation and the problem formulation. We have concentrated our attention on four basic "failure modes":

1. State step

$$x(k+1) = \Phi(k+1, k)x(k) + w(k) + \sigma_{k+1,\theta}{}^{\nu}$$

$$z(k) = H(k)x(k) + v(k)$$

2. State jump

$$x(k+1) = \Phi(k+1, k)x(k) + w(k) + \delta_{k+1,\theta}{}^{\nu}$$

$$z(k) = H(k)x(k) + v(k)$$

3. Sensor step

$$x(k+1) = \Phi(k+1, k)x(k) + w(k)$$

$$z(k) = H(k)x(k) + v(k) + \sigma_{k,\theta}{}^{\nu}$$

4. Sensor jump

$$x(k+1) = \Phi(k+1, k)x(k) + w(k)$$

$$z(k) = H(k)x(k) + v(k) + \delta_{k,\theta}{}^{\nu}$$

We have proposed a number of other failure models, but have concentrated on these four, since they provide a simple and analytically tractable framework for this basic study and also since it is felt that detectors based on these models should be able to detect other failures (such as "off failures").

Recall that the GLR approach involves the implementation of a Kalman filter based on a "no failure" assumption. In this case the filter innovations take the form

$$\gamma(k) = G_i(k; \theta)\nu + \tilde{\gamma}(k)$$

where $\tilde{\gamma}$ is the residual if there is no failure, ν is the failure magnitude, θ the failure time, and i the failure mode. The precomputable matrix $G_i(k; \theta)$ is called the failure signature and characterizes the way in which a failure of type i propagates through the system and filter. The GLR system examines the residuals, determines if there is a failure, and then estimates the time and magnitude of the failure, as well as deciding on failure type. In order to keep the detector computationally tractable, we search over a "window of residuals" -- i.e. we restrict our estimate of θ to lie in the range

$$k - M \leq \theta \leq k - N$$

A slightly simplified version of GLR is simplified GLR (SGLR) in which one hypothesizes a value for ν, thus avoiding the problem of estimation of ν. The utility of this approach is that it is quite similar in performance characteristics to GLR, it requires less computation, and it is more readily analyzed than full GLR (see the next section).

In the previous report, we developed the basic GLR and SGLR equations

for all four failure modes, and described the first steps in developing a GLR computer package. In this report we describe progress along several lines. Section II deals with several analytical tools that have been and are being developed in order to gain insight into the workings of GLR and also to provide some tools for the inevitable design tradeoff studies. We consider the usual probabilities of false alarm and correct detection, but we also define and consider several other probabilities of interest. One of these is called cross detection probability, and it represents a measure of the indistinguishability of different failure modes. The other is wrong time detection (i.e. detecting a failure at the incorrect time). This is a useful piece of information in evaluating the overall performance of GLR, since, if some of the wrong time probabilities are large, one can improve overall detector performance by examining a window of values for θ. A number of issues involving these performance measures are discussed in Section II.

In Section III we describe the test problem used in our studies. We have used a second order, simplified model of the longitudinal dynamics of the F-8 aircraft. In Section IV we discuss the application of the performance tools of Section II to the test problem, and in Section V we describe the results of a set of simulation runs. We have attempted in this latter section to describe the qualitative behavior of detector performance. Section VI contains a description of the GLR computer package that has been developed. The present package allows one to perform a variety of analytical tests (Sec. II) and to simulate system performance. One can run several types of detectors simultaneously, thus allowing a study of cross-detection behavior. In addition, one can design detectors based on one system model

and can simulate its performance when the real system is different. This option will allow us to study the robustness of the detector.

We note that the presentation given here is somewhat unpolished as it represents a status and not a final report. Thus, there are numerous loose ends and open questions throughout the report. We have collected these in Section VII in which we describe the next tasks to be undertaken. More finished descriptions of our research will be forthcoming (specifically, in the theses of Chow and Bueno).

II. Performance measures and probability computations

In this section, we report our efforts in studying the performance of the GLR technique. We define the probabilities of correct detection, false alarm, cross detection and wrong time as some measures of performance. Since these quantities provide incomplete evaluation of the detection scheme, their significance and limitations are discussed. It is shown that these probabilities require the evaluation of chi squared and gaussian integrals for the full GLR and simplified GLR respectively. Computational algorithms for such probabilities are presented. As an example of applying these performance measures, probabilities of correct detections and false alarms for the second order model of the F-8 aircraft are considered in Section IV. Such analytical results will be verified by the simulation studies described in a later section (Section V).

The probabilities of correct detection (P_D), false alarm (P_F), cross detection ($P_{i/j}$) and wrong time (P_{θ/θ_t}) are defined as follows.

$$P_D = \text{Prob}\ (\ell(k;\theta) > \varepsilon | \alpha = \beta,\ \nu,\ \theta = \theta_t)$$

$$P_F = \text{Prob}\ (\ell(k;\theta) > \varepsilon | \alpha \neq 0,\ \beta = 0,\ \nu = 0)$$

$$P_{i/j} = \text{Prob}\ (\ell(k;\theta) > \varepsilon | \alpha = i,\ \beta = j,\ \alpha \neq \beta,\ \nu,\ \theta = \theta_t)$$

$$P_{\theta/\theta_t} = \text{Prob}\ (\ell(k;\theta) > \varepsilon | \alpha = \beta,\ \nu,\ \theta \neq \theta_t)$$

where α denotes the failure mode that the GLR detector is based upon and β denotes the type of failure that actually occurs. Both α and β may take the values 1, 2, 3, or 4 representing the four modes of failures. The value of 0 for β is used for the case that no failure occurs. Also, θ_t is the true time of failure; θ is the hypothesized time of failure; ν is the true failure vector; and ε is the threshold.

There are many aspects to the evaluation of a detection scheme and a single index is not sufficient to indicate the quality of the scheme. The above probabilities are some convenient quantities defined in order to study some aspects of the GLR detector performance. P_D is the confidence that one would have in the detector since it is the probability of detecting a failure when a failure actually occurred. P_F measures the negative quality of the detector as it is the probability that a failure is signaled while no failure has occurred. Both $P_{i/j}$ and P_{θ/θ_t} are more subtle measures of performance, since they pertain to the ability of the detector to distinguish failures of different types and different failure times respectively.

Note that these probabilities are defined at each point in time assuming no knowledge of the $\ell(k; \theta)$ at other times. It is clear that $\ell(k; \theta)$ and $\ell(j; \phi)$ are correlated whenever the intervals $[\theta, k]$ and $[\phi, j]$ overlap. Since the GLR detector operates over ranges of values of k and θ (both as real time and as hypothesized failure time vary), a better set of performance measures might be "interval" versions of the probabilities defined earlier. For example, one might be interested in determining probabilities such as

$$P_D^* = \text{Prob}\left(\ell(k;\theta) > \varepsilon \,|\, \alpha = \beta,\ \nu,\ \theta = \theta_t,\ \ell(j;\theta) < \varepsilon,\ \theta \leq j < k\right)$$

This is the probability that we will <u>first</u> detect the failure at time k and is extremely useful in evaluating delay time in detection.

The modified probabilities require the joint densities of $\ell(k; \theta)$ and $\ell(j; \phi)$ which are difficult to compute in the full GLR case since correlated noncentral Chi squared (χ^2) random variables are involved.

However, in the simplified GLR case, $\ell(k;\theta)$ and $\ell(j;\phi)$ are jointly distributed gaussian random variables, and the study of the modified probabilities is easier in this case and hopefully will lead to a better understanding of the full GLR. This study will be included in the next report.

For the computation of the probabilities defined at the beginning of this section, the density functions of $\ell(k;\theta)$ under the stated conditions are required. It is shown in II.1 that the full GLR is a noncentral χ^2 random variable while the simplified GLR is a gaussian random variable (II.2). The computation of noncentral χ^2 probabilities is considered in II.3. The noncentrality parameter (δ^2) of the χ^2 density and the mean of the guassian density of GLR systems reflect the effect of the failures on the $\ell(k;\theta)$. In II.4, these parameters are examined under the condition of correct detection.

II.1 Full GLR probability density

Consider a detector that hypothesizes a type i failure with failure time = θ while an actual failure ν of type j occurred at θ_t. The actual residuals and GLR outputs then are given by

$$\gamma(k) = \tilde{\gamma}(k) + G_j(k;\theta_t)\nu$$

$$d(k;\theta) = \sum_{m=\theta}^{k} G_i'(m;\theta)V^{-1}(m)\gamma(m)$$

$$= \sum_{m=\theta}^{k} G_i'(m;\theta)\,V^{-1}(m)\,[\tilde{\gamma}(m) + G_j(m;\theta_t)\nu]$$

$$\ell(k;\theta) = d'(k;\theta)\,C_{i/i}^{-1}(k;\theta/\theta)\,d(k;\theta)$$

where

$G_j(k;\ \theta)$ is the G matrix corresponding to a type j failure, $\tilde{\gamma}(k)$ is the unbiased, white part of the residual, and

$$C_{i/i}(k;\ \theta/\theta) \triangleq \sum_{m=\theta}^{k} G_i'(m;\ \theta)V^{-1}(m)\ G_i(m;\ \theta)$$

Note that $C_{i/i}(k;\ \theta/\theta) = C(k;\ \theta)$ of a type i detector.

Since $V^{-1}(m)$ is a positive definite symmetrix matrix, $C_{i/i}(k;\ \theta/\theta)$ is positive semi-definite symmetric matrix. Then there exist an orthonormal matrix T such that

$$\Lambda_{i/i}(k;\ \theta/\theta) = T^{-1}C_{i/i}(k;\ \theta/\theta)T$$

where $\Lambda_{i/i}(k;\ \theta/\theta)$ is a diagonal matrix and the diagonal elements are the eigenvalues $\lambda_1, \lambda_2 \cdots \lambda_n$ of $C_{i/i}(k;\ \theta/\theta)$ (n is the dimension of $C_{i/i}(k;\ \theta/\theta)$). Assuming $C^{-1}_{i/i}(k;\ \theta/\theta)$ exists, define

$$\ell(k;\ \theta) = \{d'(k;\ \theta)T\}\{T^{-1}C^{-1}_{i/i}(k;\theta/\theta)T\}\{T^{-1}d(k;\ \theta)\}$$

$$\triangleq v'(k;\ \theta)\ \Lambda^{-1}_{i/i}(k;\ \theta_1)\ \ v(k;\ \theta_2)$$

Then $v(k;\ \theta)$ is a guassian random vector:

$$v(k;\ \theta) = T' \sum_{m=\theta}^{k} G_i'(m;\ \theta)V^{-1}(m)[\tilde{\gamma}(m) + G_j(m;\ \theta_t)\nu]$$

$$E\{v(k;\ \theta)\} = T' \sum_{m=\theta}^{k} G_i'(m;\theta)V^{-1}(m)G_j(m;\ \theta_t)\nu$$

$$\triangleq T'C_{i/j}(k;\ \theta/\theta_t)\nu$$

$$E\{v(k;\ \theta)v'(k;\ \theta)\}$$

$$= T'C_{i/i}(k;\theta/\theta)T + T'C_{i/j}(k;\theta/\theta_t)\nu\nu^T C'_{i/j}(k;\theta/\theta_t)\}T$$

$$= \Lambda_{i/i}(k;\ \theta/\theta) + [E\{v(k;\theta)\}][E\{v(k;\theta)\}]'$$

Hence $\Lambda_{i/i}(k;\ 0/\theta)$ is the covariance of $v(k;\ \theta)$. Since $\Lambda_{i/i}(k;\ 0/\theta)$ is diagonal, elements of $v(k;\ \theta)$ are independent of one another. $\ell(k;\ \theta)$ can be expressed as the summation:

$$\ell(k;\ \theta) = \sum_{m=1}^{n} \frac{v_m^2\ (k;\ \theta)}{\lambda_m}$$

where $v_m(k;\ \theta)$ is the m^{th} component of $v(k;\ \theta)$. Then each term in the above summation is the square of a gaussian random variable with unit variance and mean of $\frac{\bar{v}_m^2\ (k;\ \theta)}{\lambda_m}$ ($\bar{v}_m(k;\ \theta)$ is the mean of $v_m(k;\ \theta)$). Therefore, $\ell(k;\ \theta)$ is a noncentral χ^2 random variable with n degrees of freedom. The noncentrality parameter (δ^2) can be shown to be

$$\delta^2 = [E\{v(k;\ \theta)\}]'\Lambda^{-1}_{i/i}\ (k;\ \theta/\theta)[E\{v(k;\ \theta)\}]$$

$$= \nu\ C'_{i/j}(k;\ \theta/\theta_t)\ C^{-1}_{i/i}\ (k;\ \theta/\theta)\ C_{i/j}(k;\ \theta/\theta_t)\nu$$

Note that no assumption is made on i, j, θ and θ_t. The derivation includes the conditions defining P_D, P_F, $P_{i/j}$ and P_{θ/θ_t} as special cases as well as others which are not considered presently. In any event, $\ell(k;\ \theta)$ is a noncentral χ^2 random variable with n degrees of freedom and δ^2 dependent on the conditions hypothesized. Specializing to the four cases of current interest, we have,

(1) P_D: $\theta = \theta_t$, $i = j$

$$\delta^2 = \nu'\ C_{i/i}(k;\theta/\theta)\nu$$

(2) P_F: $i = j$, $\nu = 0$

$$\delta^2 = 0$$

$\ell(k;\ \theta)$ becomes a central χ^2 random variable

(3) $P_{i/j}$: $i \neq j$, $\theta = \theta_t$

$$\delta^2 = \nu' C'_{i/j}(k; \theta/\theta) C^{-1}_{i/i}(k; \theta) C_{i/j}(k; \theta/\theta)$$

(4) P_{θ/θ_t}: $i = j$ $\theta \neq \theta_t$

Note that the different relationships among θ, θ_t, k have different physical meanings, for instance,

$k < \theta_t \leq \theta$
$k < \theta \leq \theta_t$
$\theta_t \leq k < \theta$ } not meaningful

$\theta \leq k < \theta_t$ false alarm

$\theta_t < \theta \leq k$
$\theta < \theta_t \leq k$ } wrong time

then

$$\delta^2 = \nu' C'_{i/i}(k; \theta/\theta_t) C_{i/i}(k; \theta/\theta) C_{i/i}(k; \theta/\theta_t) \nu$$

The probabilities (P_D, P_F, $P_{i/j}$, P_{θ/θ_t}) are simply the integral of the density functions of $\ell(k; \theta)$ from $\ell = \varepsilon$ to $\ell = +\infty$.

II.2 Simplified GLR probability density

Consider a simplified GLR (SGLR) detector set to detect a failure ν_o of type i with failing time = θ while a true failure ν of type j actually occurred at θ_t.

$$\gamma(k) = \tilde{\gamma}(k) + G_j(k; \theta_t)\nu$$

$$\ell(k;\ 0) = \sum_{m=0}^{k} [2\gamma(m) - G_i(m;\ \theta)\nu_o]' V^{-1}(m) G_i(m;\ \theta)\nu_o$$

$$= \sum_{m=\theta}^{k} 2\tilde{\gamma}'(m) V^{-1}(m) G_i(m;\ \theta)\ \nu_o$$

$$+ 2 \sum_{m=\theta}^{k} \nu' G_j'(m;\ \theta_t) V^{-1}(m) G_i(m;\ \theta)\ \nu_o$$

$$- \nu_o' \sum_{m=\theta}^{k} G_i'(m;\ \theta) V^{-1}(m) G_i(m;\ \theta)\nu_o$$

$$= \sum_{m=\theta}^{k} 2\ \nu_o'\ G_i'(m;\ \theta) V^{-1}(m)\tilde{\gamma}(m)$$

$$+ 2\ \nu_o'\ C_{i/j}(k;\ \theta/\theta_t)\nu - \nu_o'\ C_{i/i}(k;\ \theta/\theta)\nu_o$$

Since $\tilde{\gamma}(m)$ are zero mean, independent gaussian random vectors, $\ell(k;\ \theta)$ is a gaussian random variable with mean $(\bar{\ell})$ and variance (σ^2):

$$\bar{\ell} = E\{\ell(k;\ \theta)\} = 2\nu_o'\ C_{i/j}(k;\ \theta/\theta_t)\nu - \nu_o'\ C_{i/i}(k;\ \theta/\theta)\nu_o$$

$$\sigma^2 = E\{[\ell(k;\ \theta) - \bar{\ell}]^2\}$$

$$= 4\ \nu_o' \sum_{m=\theta}^{k} G_i'(m;\ \theta) V^{-1}(m) G_i(m;\ \theta)\ \nu_o$$

$$= 4\ \nu_o'\ C_{i/i}(k;\ \theta)\ \nu_o$$

Note that the variance is the same for all cases whereas the mean varies. For the four cases of interest:

(1) P_D: $i = j$, $\theta = \theta_t$, $\nu = \nu_o$

$$\bar{\ell} = \nu'\ C_{i/i}(k;\ \theta/\theta)\nu$$

(2) P_F: $i = j \quad \nu = 0$

$$\bar{\ell} = -\nu_o' C_{i/i}(k;\theta/\theta)\nu_o$$

(3) $P_{i/j}$: $i \neq j$, $\nu = \nu_o$, $\theta = \theta_t$

$$\bar{\ell} = 2\nu_o' C_{i/j}(k;\theta/\theta) \cdot \nu_o' C_{i/i}(k;\theta/\theta)\nu_o$$

(4) P_{θ/θ_t}: $i = j$, $\nu = \nu_o$,

$\theta < \theta_t < k$ or

$\theta_t < \theta < k$

$$\bar{\ell} = 2\nu_o' C_{i/i}(k;\theta/\theta_t)\nu - \nu_o' C_{i/i}(k;\theta/\theta)\nu_o$$

Another probability of cross detection $P_{i/i}(\nu)$ may be defined for the simplified GLR.

$$P_{i/i}(\nu): i = j, \theta = \theta_t, \nu \neq \nu_o$$

$$\bar{\ell} = \nu_o' C_{i/i}(k;\theta/\theta)[2\nu - \nu_o]$$

Simplified GLR is polarized to detect a special failure direction. $P_{i/i}(\nu)$ provides a measure of the ability of simplified GLR to detect <u>other failure directions</u> than the hypothesized one (ν_o). This quantity can also be used as a measure of the distinguishability of different failure directions for a simplified GLR detector. Given the basic similarity of the GLR and SGLR algorithms, these calculations should shed light on the properties of full GLR.

The desired probabilities are easily obtained by integrating gaussian distributions.

II.3 The χ^2 random variable

A. The central χ^2 random variable u with n degrees of freedom is the sum of squares of n independent, zero mean, unit variance gaussian random variables or more precisely,

$$u \quad \sum_{i=1}^{n} x_i^2$$

when $x_i \sim N(0, 1)$

and $E\{x_i x_j\} = 0, \quad i \neq j.$

Then the density function of u is:

$$f_u^n(u) = \begin{cases} \dfrac{1}{2^{n/2}\,\Gamma(\frac{1}{2}n)}\, e^{-u/2}\, u^{\frac{n}{2}-1} & u > 0 \\ 0 & u \leq 0 \end{cases}$$

where $\Gamma(\cdot)$ is the gamma function.

There is a FORTRAN subroutine (CDTR) in the IBM Scientific Subroutine Package that can be readily used to compute the integral of the above density, i.e. the quantity

$$P_u^n(u \leq \varepsilon) = \int_{-\infty}^{\varepsilon} f_u^n(u)du = \int_0^{\varepsilon} f_u^n(u)du$$

Then the false alarm probability of a detector set to detect a failure in an n-dimensional system is

$$P_F = 1 - P_u^n(u \leq \varepsilon).$$

B. The noncentral χ^2 random variable ω with n degrees of freedom is the sum of squares of n independent, nonzero mean, unit variance gaussian random variables with the noncentrality parameter defined as

$$\delta^2 = \sum_{i=1}^{n} [E(x_i)]^2$$

With $\delta^2 = 0$, ω is a central χ^2. The density for ω is

$$f^n_{\omega,\delta^2}(\omega) = \begin{cases} \dfrac{1}{\sqrt{\pi}\, 2^{n/2}}\, e^{-\frac{1}{2}(\delta^2+\omega)}\, \omega^{\frac{n-2}{2}} \displaystyle\sum_{j=0}^{\infty} \frac{(\delta^2\omega)^j\, \Gamma(j+\frac{1}{2})}{(2j)!\, \Gamma(j+\frac{1}{2}\, n)} & \omega > 0 \\ 0 & \omega \leq 0 \end{cases}$$

Recall

$$\Gamma(\tfrac{1}{2}) = \sqrt{\pi}$$

Then

$$f^n_{\omega,\delta^2}(\omega) = e^{-\frac{1}{2}\delta^2} \sum_{j=0}^{\infty} (\delta^2)^j\, \omega^{\frac{n-2}{2}+j}\, e^{-\frac{1}{2}\omega}\, \frac{\Gamma(j+\frac{1}{2})}{2^{n/2}\, \Gamma(\frac{1}{2})\, (2j)!\, \Gamma(\frac{1}{2}\, n+j)} \qquad \omega > 0$$

$$= e^{-\frac{1}{2}\delta^2} \sum_{j=0}^{\infty} (\delta^2)^j\, \omega^{\frac{n}{2}+j-1}\, e^{-\frac{1}{2}\omega}\, \frac{(j-1+\frac{1}{2})(j-2+\frac{1}{2}) \cdots \frac{1}{2}\, \Gamma(\frac{1}{2})}{2^{n/2}(2j)!\, \Gamma(\frac{1}{2})\, \Gamma(j+\frac{1}{2}\, n)}$$

$$= e^{-\frac{1}{2}\delta^2} \sum_{j=0}^{\infty} (\delta^2)^j \omega^{\frac{n}{2}+j-1}\, e^{-\frac{1}{2}\omega}\, \frac{2^{-j}(2j-1)(2j-3)\cdots(3)(1)\Gamma(\frac{1}{2})}{2^{\frac{n}{2}}(2j)(2j-1)(2j-2)(2j-3)\cdots(3)(2)(1)\Gamma(\frac{1}{2})\Gamma(j+\frac{1}{2}\, n)}$$

$$= e^{-\frac{1}{2}\delta^2} \sum_{j=0}^{\infty} (\delta^2)^j\, \frac{1}{j!\, 2^{n/2+2j}\, \Gamma(j+\frac{1}{2}\, n)}\, \omega^{\frac{n}{2}+j-1}\, e^{-\frac{1}{2}\omega}$$

$$= e^{-\frac{1}{2}\delta^2} \sum_{j=0}^{\infty} \frac{(\delta^2)^j}{2^j j!}\, f_u^{n+2j}(\omega)$$

hence,

$$p^n_{\omega,\delta^2}(\omega \leq \varepsilon) = e^{-\frac{1}{2}\delta^2} \sum_{j=0}^{\infty} \frac{(\delta^2/2)^j}{j!}\, p_u^{n+2j}(u \leq \varepsilon)$$

$p_u^{n+2j}(u \leq \varepsilon)$ may be computed using the IBM subroutine CDTR. Then

$P^n_{\omega,\delta^2}(\omega \leq \varepsilon)$ is calculated by performing the summation. The infinite sequence in the expression of $P^n_{\omega,\delta^2}(\omega \leq \varepsilon)$ can be proved to be convergent. For wide ranges of δ^2 and ε, the limit of the series is effectively attained by summing less than thirty terms. We also note that

$$\lim_{\varepsilon \to \infty} P^n_{\omega,\delta^2}(\omega \leq \varepsilon) = 1 \quad \text{for all } \delta^2 \geq 0$$

$$\lim_{\delta^2 \to \infty} P^n_{\omega,\delta^2}(\omega \leq \varepsilon) = 0. \quad 0 \leq \varepsilon < \infty$$

Hence, P_D, $P_{i/j}$ and P_{θ/θ_t} are increasing functions of δ^2 and approach 1 as δ^2 goes to ∞ for any finite value of ε, the threshold (see Figure 2.1).

II.4 A study of the probability of correct detection (P_D)

Recall for full GLR,

$$P_D = \text{Prob}(\ell(k;\theta) > \varepsilon | \alpha = \beta = i, \quad \theta = \theta_t)$$

Under this condition, $\ell(k;\theta)$ is a noncentral χ^2 random variable with noncentrality parameter δ^2.

$$\delta^2 = \nu' C_{i/i}(k;\theta/\theta)\nu$$

By the definition

$$C_{i/i}(k;\theta/\theta) \triangleq C(k;\theta) \text{ of the type i detector}$$

Then

$$\delta^2 = \nu' C(k;\theta)\nu$$

For simplified GLR,

$$P_D = \text{Prob}(\ell(k;\theta) > \varepsilon | \alpha = \beta = i, \theta = \theta_t, \nu = \nu_c)$$

$\ell(k;\theta)$ is then a gaussian random variable with mean $(\bar{\ell})$ and variance (σ^2).

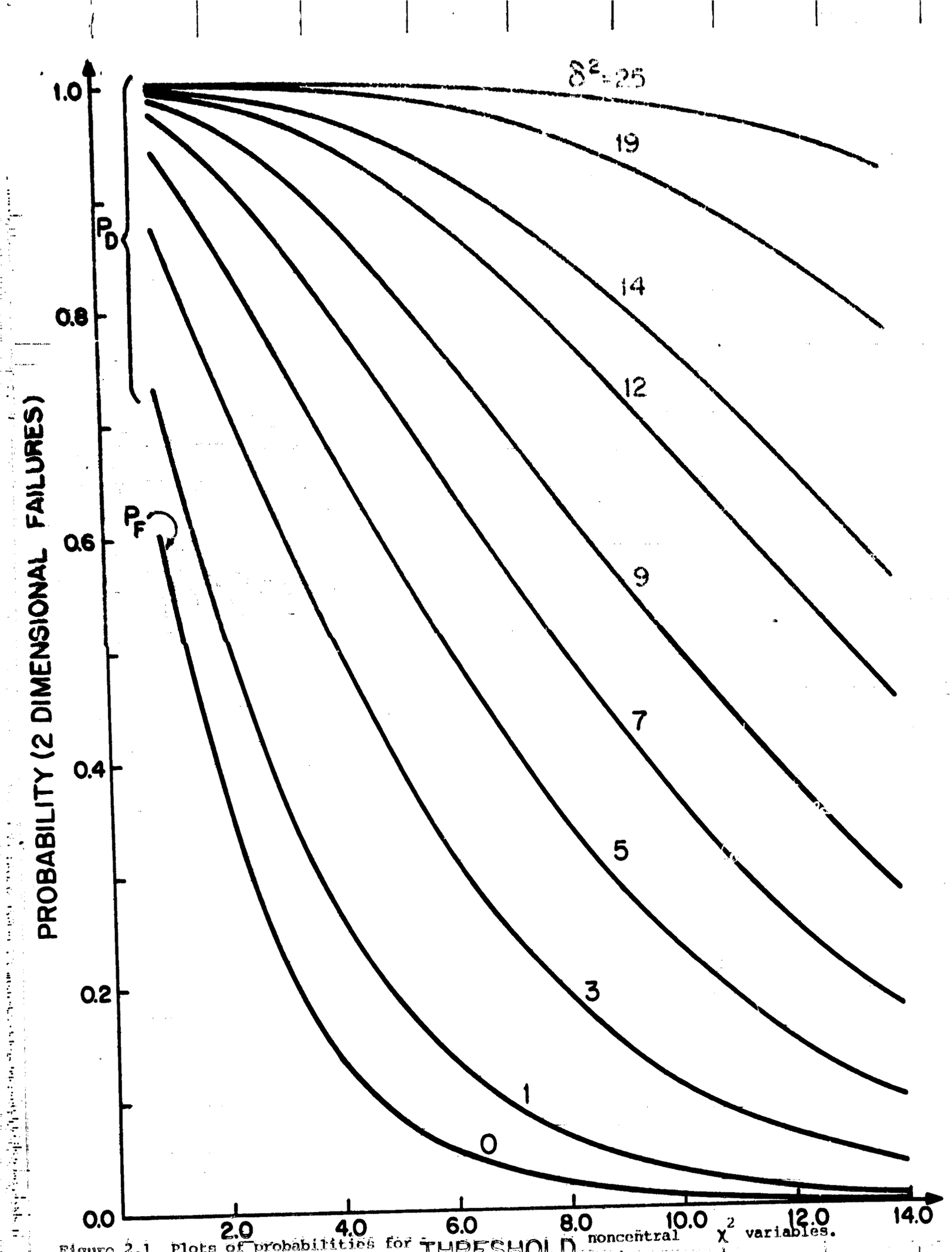


Figure 2.1 Plots of probabilities for noncentral χ^2 variables.

$$\bar{\ell} = \nu' C(k;\ 0)\nu$$

$$\sigma^2 = 4\ \bar{\ell}$$

In both cases, the P_D's are increasing functions of δ^2 and $\bar{\ell}$ which evolve with time (k). An understanding of the evolution of the P_D's requires the evolution of δ^2 and $\bar{\ell}$ which in trun require the behavior of $C(k;\ \theta)$ as a function of time.

In the following, we present an analysis of the behavior of $C(k;\ \theta)$ for time invariant systems.

In a time invariant system and ste dy-state Kalman filter, $C(k;\ \theta)$ becomes dependent on the difference between the true failing time and observation time $(k-\theta)$. For convenience, we let $r = k - \theta$. The four different types of detector are considered separately.

1. State Jump Detector

$$F(r) = \sum_{j=0}^{r} \Theta^{r-j}\ KH\ \Phi^{j}$$

where $\Theta = [I-KH]\Phi$; K is the steady state Kalman gain, Φ is the system matrix and H is the observation matrix. Both the system and the filter are assumed to be stable. Then the magnitude of the eigenvalues of Φ and Θ is strictly less than 1, i.e.

$$|\lambda_i(\Phi)| < 1 \qquad i = 1,\ 2,\ \ldots\ n$$

$$|\lambda_i(\Theta)| < 1 \qquad i = 1,\ 2,\ \ldots\ n$$

where n is the dimension of Φ and Θ. Consider the norm $||\cdot||$ of a nxm matrix A,

$$||A|| = \max_{||x||=1} (x'A'Ax)^{1/2}$$

where x is a m-vector.

For a square matrix Λ with all eigenvalues of magnitudes less than 1, it can be shown that $||A|| < 1$.

For a jump in the state,

$$F(r) = \sum_{j=0}^{r} \Theta^{r-j} KH\Phi^{j}$$

$$||F(r)|| \leq \sum_{j=0}^{r} ||\Theta^{r-j}KH\Phi^{j}|| \leq \sum_{j=0}^{r} ||KH||\ \rho^{r} = ||KH||(r+1)\ \rho^{r}$$

where $\rho = \max \{||\Phi||, ||\Theta||\}$

Since $\rho < 1$, there exist a $\alpha > 0$ such that $\rho = e^{-\alpha}$. Then

$$||F(r)|| \leq ||KH||\ (r+1)\ e^{-\alpha r}$$

The RHS goes to zero as $r \to \infty$. Therefore

$$\lim_{r \to \infty} F(r) = 0$$

Similarly, for G(r),

$$||G(r)|| = ||H[\Phi^{r} - \Phi F(r-1)]||$$

$$\leq ||H||[||\Phi||^{r} + ||\Phi||\ ||F(r-1)||]$$

$$\leq ||H||\ [\rho^{r} + ||KH||\ r\ \rho^{r}]$$

Hence G(r) also approaches zero as $r \to \infty$. Now consider C(r). Define

$$\Delta C(r,s) \triangleq C(r) - C(s) \qquad , r < s$$

$$= \sum_{j=r+1}^{s} G'(j)\ v^{-1} G(j)$$

$$||\Delta C(r,s)|| \leq \sum_{j=r+1}^{s} ||v^{-1}||\ ||H||^2\ [\rho^j + ||KH||\ j\rho^j]$$

$$\leq ||v^{-1}||\ ||H||^2\ [(s-r)\rho^r + ||KH||\ (s-r) + \rho^r]$$

As $r \to \infty$, the terms in the bracket approach 0. Hence

$$\lim_{r \to \infty} ||\Delta C(r,s)|| = 0 \qquad r < s$$

This shows the $\{C(1), C(2), \ldots C(r) \ldots\}$ is a Cauchy sequence and hence converges to a finite constant matrix. Noting that C has the interpretation as the information matrix associated with estimating ν, we see that there is a finite amount of information concerning ν in the residuals γ (this is clear since both the system and filter are stable and the failure is a transient effect -- i.e., a jump). By determining the rate of convergence of $C(r)$, we can choose a waiting time r^* such that there is essentially no information in $\gamma(k)$ concerning failures at time θ, where $k - \theta > r^*$.

2. Step in the state

$$F(r) = \sum_{i=0}^{r} \sum_{j=i}^{r} \Theta^{r-j}\ KH\Phi^{j-i}$$

$$= \sum_{j=0}^{r} \Theta^{r-j} \sum_{i=0}^{j} KH\Phi^{i}$$

$$= \sum_{j=0}^{r} \Theta^{r-j}\ KH\ [I - \Phi^{j+1}][I - \Phi]^{-1}$$

$$= \sum_{k=0}^{r} \Theta^{r-j}\ KH[I - \Phi]^{-1} - \sum_{j=0}^{r} \Theta^{r-j}\ KH\Phi^{j+1}[I - \Phi]^{-1}$$

$$= [I - \Theta^{r+1}][I - \Theta]^{-1}KH[I - \Phi]^{-1} - [\sum_{j=0}^{\ell} \Theta^{r-j}KH\Phi^{j}]\Phi[I - \Phi]^{-1}$$

As $r \to \infty$, the first term becomes $[I - \Theta]^{-1}KH[i - \Phi]^{-1}$ and the second goes to 0 following the reasoning for the state jump case. $[I - \Theta]^{-1}$ and $[I - \Phi]^{-1}$ exist because $|\lambda_i(\Theta)| < 1$, $|\lambda_i(\Phi)| < 1$ for $i = 1, 2, \ldots n$.

$$G(r) = H\left[\sum_{j=0}^{r} \Phi^{r-j} - \Phi F(r-1)\right]$$

$$= H[I - \Phi^{r+1}][I - \Phi]^{-1} - H\Phi F(r-1)$$

As $r \to \infty$, the first term becomes $H[I \quad \Phi]^{-1}$ and the second, $H\Phi[I-\Theta]^{-1}KH[I-\Phi]^{-1}$. Hence G(r) reaches a constant as $r \to \infty$. $G'(j)v^{-1}G(j)$ is positive semi-definite and attains a steady state value $G'(\infty)v^{-1}G(\infty)$. Thus it is possible that some of the eigenvalues of C(r) grow as r increases indicating that some failure vectors w .1 cause a growing δ^2. Therefore, an actual failure vector of this nature will cause P_D to approach 1 as the waiting time (r) increases. By examining the eigenvectors and eigenvalues of C, we can determine those step failures that can be detected with arbitrarily high probability if we are willing to wait long enough.

3. Jump in sensor

$$\lim_{r \to \infty} F(r) = \lim_{r \to \infty} \Theta^r K = 0$$

$$\lim_{r \to \infty} G(r) = \lim_{r \to \infty} -H\Phi F(r-1) = 0$$

Hence C(r) for sensor jump failures behaves much like that of state jump failure.

4. Step in sensor

$$\lim_{r \to \infty} F(r) = \lim_{r \to \infty} \sum_{j=0}^{r} \Theta^j K = [I - \Theta]^{-1}K$$

$$\lim_{r \to \infty} G(r) = \lim_{r \to \infty} [I - H\Phi F(r-1)]$$

$$= I - H\Phi[I - \Theta]^{-1}K$$

Hence C(r) for sensor step failure behaves like that of state step failures.

In general, C(r) is a sum of positive semi-definite matrices and consequently may be positive definite or semi-definie matrix. If C(r) is positive semi-definite, then there are failure vectors such that the resulting δ^2 is zero, implying the failure direction cannot be seen by the detector and that certain failure directions are indistinguishable. Intuitively, one would suspect the cause for this is that this failure direction is not observable. This is true in fact. In the following, conditions for the positive definiteness of C(r) of different detectors are derived.

C(r) may be written as

$$C(r) = \left[G'(0) \vdots G'(1) \vdots \cdots \vdots G(r)\right] \begin{bmatrix} V^{-1} & & 0 \\ & \ddots & \\ 0 & & V^{-1} \end{bmatrix} \begin{bmatrix} G(0) \\ \cdots \\ G(1) \\ \cdots \\ \vdots \\ \cdots \\ G(r) \end{bmatrix}$$

$$\triangleq G'(r)\, V^{-1}\, G(r)$$

Since $V^{-1} > 0$, $V^{-1} > 0$. From the theory of linear algebra, C(r) is positive definite if the null space of G(r) is {0}. We examine G(r) for the four cases separately.

1. Jump in State

$$G(r) = \begin{bmatrix} I & & & \\ -H\Phi K & I & 0 & \\ -H\Phi\Theta K & -H\Phi K & I & \ddots \\ \vdots & \vdots & & \\ -H\Phi\Theta^{r-1}K & -H\Phi\Theta^{r-2}K & \cdots & I \end{bmatrix} \begin{bmatrix} H \\ H\Phi \\ H\Phi^2 \\ \\ H\Phi^r \end{bmatrix} \triangleq A_1(r)\, B(r)$$

$A_1(r)$ is a lower triangular matrix with identiy blocks in the diagonal. $A_1(r)$ is of full rank, and the null space of $G(r)$ is the null space of $B(r)$. Hence if the null space of $B(r)$ is $\{0\}$, $C(r)$ is positive definite. This condition on Γ ... is exactly the observability condition of the system in r steps.

2. Step in State

$$G(r) = \begin{bmatrix} I & & & 0 \\ I-H\Phi K & I & & \\ I-H\Phi \sum_{j=0}^{1} \Theta^j K & I-H\Phi K & I & \\ \vdots & & & \\ I-H\Phi \sum_{j=0}^{r-1} \Theta^j K & I-H\Phi \sum_{j=0}^{r-2} \Theta^j K & \cdots & I \end{bmatrix} \begin{bmatrix} H \\ H\Phi \\ H\Phi^2 \\ \vdots \\ H\Phi^r \end{bmatrix}$$

$$\triangleq A_2(r)\, B(r)$$

$A_2(r)$ is of full rank. $C(r)$ is positive definite if the system is observable in r steps.

3. Jump in sensor

$$G(0) = G'(0) = I$$

Hence the null space of $G(r)$ is always $\{0\}$. $C(r)$ is always positive definite in this case.

4. Step in sensor

Similar to sensor jump failures,

$$G(0) = G'(0) = I$$

$C(r)$ is always positive definite in the sensor step case.

System observability makes a state failure detector sensitive to all directions of failure by causing a nonzero δ^2. Sensor failures are directly observable and hence δ^2 for sensor failures is always nonzero. δ^2 attains finite steady state values for all jump failures, while for step failure, some failure directions may cause a constant, steady-state rate of increase in δ^2. A finite steady-state δ^2 gives a limiting value of P_D (< 1); an increasing δ^2 allows one to choose a P_D arbitrarily close to 1 by waiting long enough.

The above analysis also provides guidelines for selecting window sizes. For detecting jump failures, the window does not have to be large since excessive waiting time (large r) does not increase P_D after a certain stage. A long window is appropriate in detecting small step failures. If the system of concern is not immediately observable, i.e. the null space of the B(r) matrix becomes {0} for some $r > 0$, the detection test should not be performed until the system at θ becomes observable from k, the present time. That is, in general we will calculate $\ell(k; \theta)$ over an interval of the form

$$k - M \leq \theta \leq k - N$$

where N is chosen by observability considerations, while M is chosen by the limiting behavior of P_D and by computational considerations.

III. The Test Problem to be Considered - The Two-Dimensional Longitudinal Dynamics of the F-8

III.1 Introduction

Some simulation results have been obtained on the performance of the GLR (generalized likelihood ratio) detector for the first four failure types: jumps and steps in the state, and jumps and steps in the sensors. The detector equations for these failures were implemented with a simulation of a reduced-order (2nd order) F-8 aircraft model for a range of failure magnitudes and directions in state space.

The purpose in doing this was to get some experience with the GLR approach to failure detection. Having some sample performances of the detectors provides insight in a way that helps formulation of meaningful questions for further research on GLR failure detection.

Section III.2 states in general form what the GLR approach to failure detection is based on. Section III.3 describes the second-order model used in the simulations and section III.4 presents the steady state Kalman filter designed for that model. In section III.5 the relevant equations of the detector are shown and section III.6 describes what the different failures considered are and what they represent or model in a physical system.

III.2 Generalized Likelihood Ratio Approach

Briefly, the GLR approach is as follows. There are two kinds of hypotheses:

H_o: no failure has occurred

H_i: failure of type i has occurred.

If there are m failure modes, there are m+1 hypotheses.

The filter and controller are designed based on H_o. One can then compute the effect that the various failures considered have on the filter residuals. We then have:

$$H_o: \underline{\gamma}(k) = \tilde{\underline{\gamma}}(k)$$

$$H_i: \underline{\gamma}(k) = \tilde{\underline{\gamma}}(k) + G_i(k; \theta)\underline{\nu}$$

where

$\underline{\gamma}(k) = \underline{z}(k) - H\hat{x}(k|k-1)$, the residuals from the Kalman filter

$\underline{\nu}$ = failure vector

$G_i(k; \theta)$ = failure 'signature' matrix for the effect of a failure of type i at time θ on the residuals at time k. It is precomputable for each i.

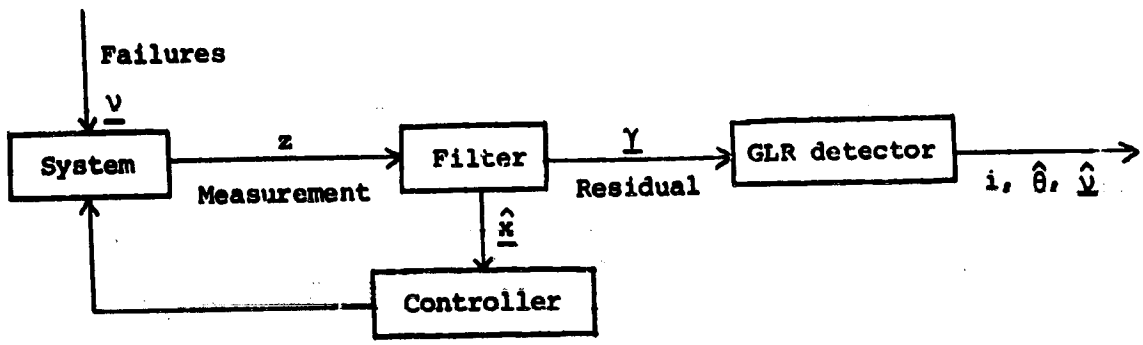
Using these computations we can perform hypothesis tests on the residuals to

i) determine if a failure has occurred

ii) identify the failure type, i

iii) estimate the size of the failure.

Schematically



III.3 System Model

The simulations were made using a second order discretized version of the longitudinal dynamics for the F-8 aircraft at flight condition 11: altitude = 20,000 ft., Mach No. = 0.6, cumulus clouds.

The motivation for using this model lies in the need to have a model of a concrete, physical system on which to try out the detectors that would provide some common grounds for comparisons. Furthermore the model provides a compromise in complexity between realism on the one hand and the amount of computation and ease of interpretation on the other. In this early phase of research on the GLR approach to failure detection some results were needed in order to understand its structure and performance characteristics. It was felt that a system of higher order would not add significantly to our understanding.

Our model is derived from the longitudinal dynamics of the F-8 linearized about flight condition 11. That model is 7-dimensional with the following state variables:

$$\underset{(7x1)}{\frac{d}{dt}\underline{x}(t)} = \underset{(7x7)(7x1)}{\underline{A}\,\underline{x}(t)} + \underset{(7x1)(1x1)}{\underline{B}\,u(t)} + \underset{(7x1)(1x1)}{\underline{L}\,\xi(t)}$$

x_1 = q, pitch rate . . . . . (rad./sec)

x_2 = v, velocity - V_o [V_o = Mach no. x speed of sound] . . . (ft/sec)

x_3 = α, angle of attack - trim value . . . . . (rad.)

x_4 = θ, pitch attitude . . . . . (rad.)

x_5 = δ_e, elevator deflection - trim value . . . . . (rad.)

x_6 = δ_{e_c}, commanded elevator angle . . . . . (rad.)

x_7 = w, normalized wind disturbance . . . . . (rad.)

The control variable is:

$$u(t) = \dot{\delta}_{e_c}(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(rad./sec)}$$

$\underline{A}$, $\underline{B}$, and $\underline{L}$ are constant matrices whose dimensions are indicated.

State variables $x_5 = \delta_e$ and $x_6 = \delta_{e_c}$ account for the dynamics of the actuators and $x_7 = w$ is the output of a first-order linear system driven by white noise. w(t) models a wind disturbance with power spectral density given by:

$$\Psi = \frac{\sigma^2}{\pi} \frac{L}{V_o} \left\{ \frac{4}{4 + (\frac{L}{V_o}\omega)^2} \right\} \qquad [\omega \neq w = x_7]$$

For flight condition 11 we have:

$L = 2{,}500$ ft.

$V_o = (0.6)(1{,}036.93 \text{ ft/sec}) = 622.150$ ft/sec

$\sigma = 15$ ft/sec (cumulus clouds)

The five sensor measurements $\underline{z}(t)$ are given by:

$$\underset{(5x1)}{\underline{z}(t)} = \underset{(5x7)}{\underline{C}} \underset{(7x1)}{\underline{x}(t)} + \underset{(5x1)}{\underline{\Theta}(t)}$$

$z_1 = z_q$, pitch rate measurement

$z_2 = z_v$, velocity error measurement

$z_3 = z_\theta$, pitch attitude measurement

$z_4 = z_{\delta_e}$, elevator angle measurement

$z_5 = z_{a_z}$, normal acceleration measurement

$\underline{C}$ is a constant (5x7) matrix and $\underline{0}$ is the vector of measurement noises, which are white and mutually independent. For more information on the model see [1].

Some of the steps taken in the reduction of the order of the model were:

- ignoring the input dynamics represented by x_5 and x_6 as they are not the main variables of interest in an aircraft dynamics model.
- eliminating x_7, the wind disturbance, as a variable and modelling its effects on the remaining ones by a white noise process.
- selecting the variables with highest signal-to-noise ratios among the observations and ignoring the rest.
- using common sense and engineering intuition to correct and/or add for any other significant interactions.

The resulting model is a two-dimensional representation of the dynamics with the new state variables:

x_1 = q, the pitch rate

x_2 = α, (angle of attack) - (trim value)

The last step was obtaining the corresponding discrete-time model in order to simplify implementation on the digital computer. The discretizing time step was T = 0.03125 sec ($\frac{1}{32}$ sec). The result,

$$\underset{(2x1)}{\underline{x}(k+1)} = \underset{(2x2)}{\underline{\Phi}}\ \underline{x}(k) + \underset{(2x2)}{\underline{G_{NX}}}\ \underline{w}(k) \qquad (1)$$

$$\underset{(2x1)}{\underline{z}(k)} = \underset{(2x2)}{\underline{H}}\ \underline{x}(k) + \underset{(2x2)}{\underline{G_{NZ}}}\ \underline{v}(k) \qquad (2)$$

where

$$E\{\underline{\omega}(k)\ \underline{\omega}(j)^T\} = \underline{I}\ \delta_{kj} \quad , \quad \delta_{kj} = \begin{cases} 0 & k \neq j \\ 1 & k = j \end{cases}$$

$$E\{\underline{v}(k)\ \underline{v}(j)^T\} = \underline{I}\ \delta_{kj}$$

$$\underline{\Phi} = \begin{bmatrix} 0.98258 & -0.14649 \\ 0.030587 & 0.97193 \end{bmatrix}$$

The eigenvalues of $\underline{\Phi}$ are:

$$\lambda_i(\underline{\Phi}) = .977 \pm j(0.0667) \quad , \quad i = 1,2$$

$$\underline{G_{NX}} = \begin{bmatrix} 0.022596 & 0.0 \\ 0.0043276 & 0.00022603 \end{bmatrix}$$

$$\underline{G_{NZ}} = \begin{bmatrix} 0.008729834 & 0.0 \\ 0.0 & 0.06 \end{bmatrix}$$

$$\underline{H} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 16.154 \end{bmatrix}$$

III.4 Filter

The filter implemented was a steady-state Kalman filter designed for the two-dimensional model under normal circumstances (hypothesis H_o):

prediction: $\hat{\underline{x}}(k|k-1) = \underline{\Phi}\ \hat{\underline{x}}\ (k-1|k-1)$

update: $\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + \underline{K}\ \underline{\gamma}\ (k)$

with

residuals: $\underline{\gamma}(k) = \underline{z}(k) - \underline{H}\ \hat{\underline{x}}\ (k|k-1)$

steady-state Kalman gain: $\underline{K} = \underline{P}(k|k-1)\ \underline{H}^T \underline{V}^{-1}$

where $\underline{P}(k|k-1) = E\{[\underline{x}(k)-\hat{\underline{x}}(k|k-1)][\underline{x}(k)-\hat{\underline{x}}(k|k-1)]^T\}$

$$\underline{V} = E\{\underline{\gamma}(k)\ \underline{\gamma}(k)^T\}$$

$$= \underline{H}\ \underline{P}(k|k-1)\underline{H}^T + \underline{G_{NZ}}\ \underline{G_{NZ}}^T$$

In this example, we have:

$$\underline{K} = \begin{bmatrix} 7.5351 \times 10^{-1} & 4.6257 \times 10^{-2} \\ 1.3527 \times 10^{-1} & 1.2748 \times 10^{-2} \end{bmatrix}$$

$$\underline{P}(k|k-1) = \begin{bmatrix} 5.6311 \times 10^{-4} & 1.0891 \times 10^{-4} \\ 1.0891 \times 10^{-4} & 2.2130 \times 10^{-5} \end{bmatrix}$$

$$V = \begin{bmatrix} 6.393264579 \times 10^{-4} & 1.759328799 \times 10^{-3} \\ 1.759328799 \times 10^{-3} & 9.374701305 \times 10^{-3} \end{bmatrix}$$

III.5 Detector

We now take a look at the detector and some of the computation involved in its implementation. For a more complete derivation see [2].

Consider a particular type of failure, i, and let k be the current time. For each $\theta \in \{k-M, \ldots, k-N\}$, corresponding to times inside a 'window' (to which we restrict the GLR to avoid a computation load which would otherwise grow indefinitely with k), we compute

$$\underline{d}(k;\ \theta) = \sum_{j=\theta}^{k} \underline{G_i}^T(j;\ \theta)\underline{V}^{-1}(j)\ \underline{\gamma}(j)$$

which then gives the likelihood ratios

$$\ell(k;\ \theta) = \underline{d}^T(k;\ \theta)\ \underline{C_i}^{-1}(k;\ \theta)\underline{d}(k;\ \theta) \quad , \quad \theta = k-M,\ k-M+1,\ \ldots,\ k-N$$

where

$$\underline{C}_i(k;\,\theta) = \sum_{j=\theta}^{k} \underline{G}_i^T(j;\,\theta)\underline{V}^{-1}(j)\underline{G}_i(j;\,\theta), \quad \theta = k-M,\ k-M+1,\ \ldots,\ k-N$$

In our case some of these quantities are time-invariant so $\underline{G}_i(k;\,\theta)$, $\underline{V}(k)$, $\underline{C}_i(k;\,\theta)$ become $\underline{G}_i(k-\theta)$, $\underline{V}$, $C_i(k-\theta)$. $\underline{C}_i(k;\,\theta)$ may be interpreted as the information matrix at time k for a failure of type i which occurred at time θ.

Detection is decided by means of the decision rule:

$$\ell(k;\,\hat{\theta}(k)) \underset{\text{no failure}}{\overset{\text{failure}}{\gtrless}} \varepsilon$$

where $\hat{\theta}(k)$ is the MLE (maximum likelihood estimate) at time k of the time of failure θ. It is taken to be the value of $\theta\varepsilon\{k-M,\ \ldots,\ k-N\}$ for which $\ell(k;\,\theta)$ is largest. The detection threshold, ε, is a design parameter to be considered in evaluating system performance. More will be said on this in section V.5.

If it is decided that there is a failure, the estimate of the failure is given by

$$\hat{\underline{\nu}}(k) = \underline{C}^{-1}(k;\,\hat{\theta}(k))\underline{d}(k;\,\hat{\theta}(k))$$

The windows used in the simulations, [k-M, k-N], had N=0 for all cases, M=10 for jump failures and M=30 for step failures.

III.6 The Failures

The failure modes considered correspond to the four types studied so far:

1) State Jumps

$$\underline{x}(k+1) = \underline{\Phi}\,\underline{x}(k) + \underline{G}_{NX}\underline{\omega}(k) + \underline{\nu}\,\delta_{k+1,\theta}$$

2) State Steps

$$\underline{x}(k+1) = \underline{\Phi}\,\underline{x}(k) + \underline{G}_{NX}\,\underline{\omega}\,(k) + \underline{\nu}\,\sigma_{k+1,\theta}$$

3) Sensor Jumps

$$\underline{z}(k) = \underline{H}\,\underline{x}(k) + \underline{G}_{NZ}\,\underline{v}(k) + \underline{\nu}\,\delta_{k,\theta}$$

4) Sensor Steps

$$\underline{z}(k) = \underline{H}\,\underline{x}(k) + \underline{G}_{NZ}\,\underline{v}(k) + \underline{\nu}\,\sigma_{k,\theta}$$

We did not include the control term in the state equations since we are not considering closed-loop systems at this point. For open-loop control nothing would change as far as the detector is concerned, since the Kalman Filter equations would incorporate the control term and its effect is cancelled.

Failures were taken in orthogonal directions in failure space, $(\nu_1,\ 0)$ and $(0,\ \nu_2)$, for a range of ν_1, ν_2 thought to be of most interest. Let us take a closer look at the failures we are considering and the situations they might model in a physical system, in our case the F-8 aircraft at the specified flight condition.

Consider an open-loop system and observation process, e.g., the one presented in section III.3 by equations (1) and (2):

$$\underline{x}(k+1) = \underline{\Phi}\,\underline{x}(k) + \underline{B}\,\underline{u} + \underline{G}_{NX}\ \ \underline{\omega} \qquad (1)$$

$$\underline{z}(k) = \underline{H}\,\underline{x}(k) + \underline{G}_{NZ}\ \ \underline{v} \qquad (2)$$

and consider the 4 basic "failure modes" described in Section I:

i) State jump

ii) State step

iii) Sensor jump

iv) Sensor step

Keeping the system equations, (1) and (2), in mind, we can say a few things about (i) - (ii). Let us examine these failures in order to provide some (albeit superficial) physical motivation for the various modes. Recall that x_1 is q and x_2 is α. Thus a state jump of the form $(\nu_1, 0)^T$ or a state step of the form $(0, \nu_2)^T$ might be used to model the effect of a sudden wind shear that leads to an increasing angle of attack. On the other hand, a jump of the form $(0, \nu_2)^T$ could be used to model a relatively long-term upward or downward gust that initially manifests itself as a shift in α. A step of the type $(\nu_1, 0)^T$ could arise from an elevator failure.

In the observation equation, cases (iii) and (iv), we have a similar situation. A failure $\underline{\nu} = (\nu_1, 0)^T$ may model a bad data point in the measurement of q in the jump case, (iii), or a permanent bias for the step case, (iv), in the same signal due to a component failure in a sensor. By analogy the same may be said about a failure $\underline{\nu} = (0, \nu_2)^T$ which then refers to the measurement of α.

Table 3.1, summarizes the failure schedule implemented in the simulations. State and sensor failure magnitudes are given in terms of σ-levels of plant and sensor noises respectively. For jump failures nothing under 1σ was looked at since such jumps would be undistinguishable from the noise. Such failure magnitudes were considered for step failures since they are detectable because their sustained presence provides more information as time passes. We will see how this is reflected in the GLR's when

we take a closer look at the $G_i(k; \theta)$ for each type of failure.

The σ's are the following:

$\begin{pmatrix} \nu_1 \\ 0 \end{pmatrix}$, pitch rate; σ_q: Plant Noise Level = 2.2596×10^{-2}

σ_q': Sensor Noise Level = 8.7298×10^{-3}

$\begin{pmatrix} 0 \\ \nu_2 \end{pmatrix}$, angle of attack; σ_α: Plant Noise Level = 4.3335×10^{-3}

σ_α': Sensor Noise Level = 6.0000×10^{-2}

DETECTOR AND FAILURE TYPE	$(\nu_1 \quad 0)^T$	$(0 \quad \nu_2)^T$
STATE JUMP	1σ, 5σ, 10σ, 20σ	1σ, 5σ, 10σ, 20σ
STATE STEP	$\frac{1}{10}\sigma$, $\frac{1}{2}\sigma$, 1σ, 5σ,	$\frac{1}{10}\sigma$, $\frac{1}{2}\sigma$, 1σ, 5σ,
SENSOR JUMP	$1\sigma'$, $5\sigma'$, $10\sigma'$, $20\sigma'$	$1\sigma'$, $5\sigma'$, $10\sigma'$, $20\sigma'$
SENSOR STEP	$\frac{1}{10}\sigma'$, $\frac{1}{2}\sigma'$, $1\sigma'$, $5\sigma'$, $10\sigma'$, $20\sigma'$	$\frac{1}{10}\sigma'$, $\frac{1}{2}\sigma'$, $1\sigma'$, $5\sigma'$, $10\sigma'$, $20\sigma'$

Table 3.1 Set of Failures Considered

IV. Application to the Second Order F-8 Model - Full GLR

The graphs in Fig. 4.1 and 4.2 represent a way of using P_D and P_F together. For a particular step failure and a threshold, the graphs provide the necessary waiting time $(k - \theta)$ for P_D to reach the values of .95 and .99. The failure size is measured in units of standard deviation of the noise (1σ = 1 standard deviation). Each threshold has a fixed value of P_F associated with it. The threshold values 5, 9, and 14 considered here have P_F's of .1, .01 and .001 respectively. For example, consider a $.1\sigma$ step failure in the pitch rate. To reach a P_D of .99 with P_F of .01, a waiting time of 36 steps is required. Each time step is 1/32 second. The waiting time may be taken as a measure of the speed of detection with a fixed rate of false alarms.

It is evident from the graph that the full GLR detector is very fast in detecting state step failure of sizes ranging from $.1\sigma$ to 5σ. For the sole purpose of detection (no estimation of ν), a small window of about 50 time steps is sufficient.

Sensor failures require much longer waiting times, as high as 6.6 hours for a $.1\sigma$ step in the angle of attack deviation sensor. This phenomenon may be explained by the nature of the associated G matrices (see Figures 5.2 and 5.3 in Section V). The entries in the G matrices are small in comparison with the G matrices of state step failure resulting in a small and slow growing δ^2. Therefore, the window of a sensor step detector should be large. Its size would depend on the sizes and direction of failure of interest.

Waiting time plots are not made for jump failures due to the following observation. For jump failures, P_D either reaches the desired

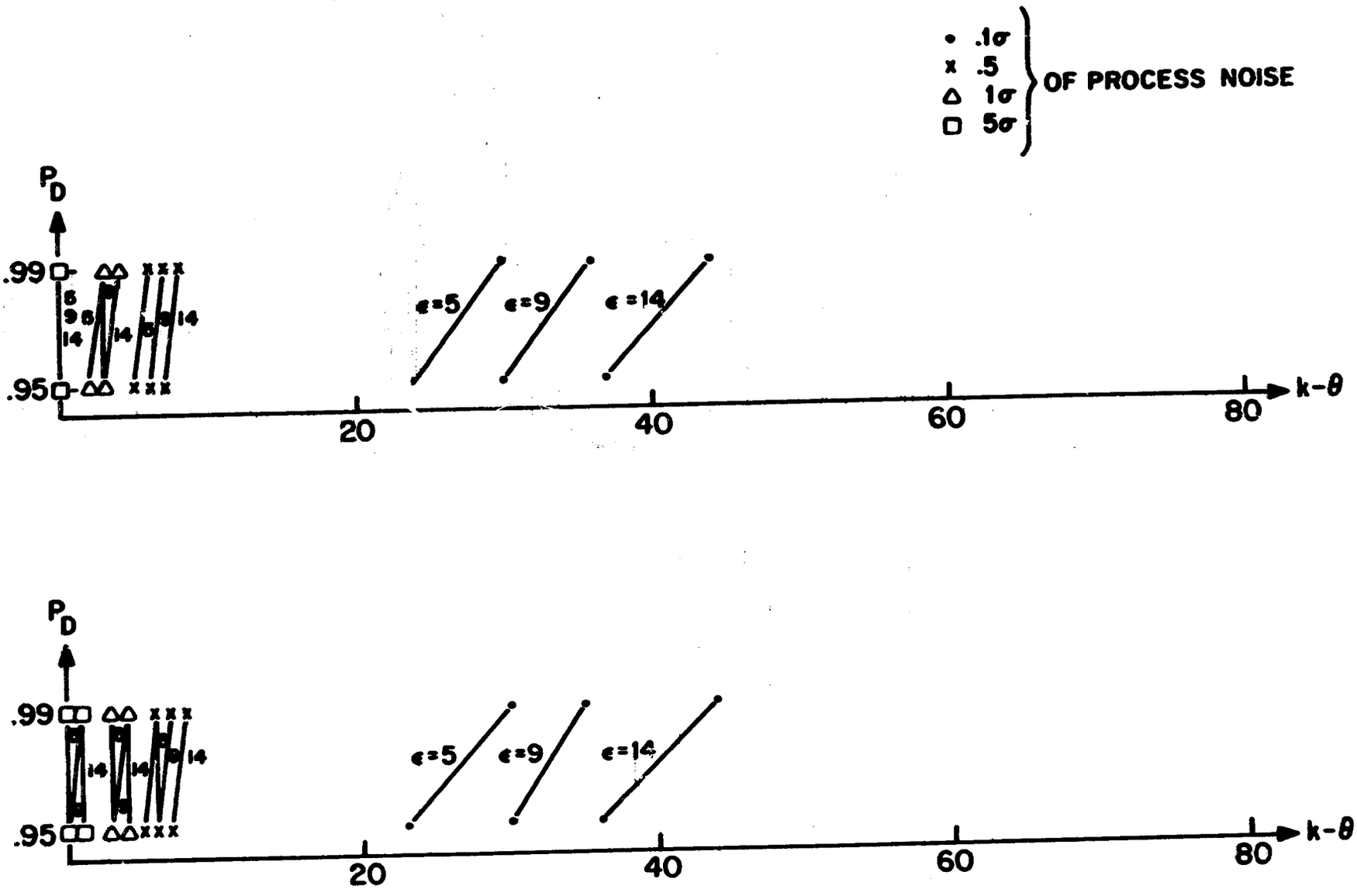


Figure 4.1 Step failure in angle of attack deviation: $\theta' = [\theta, \theta_2]$

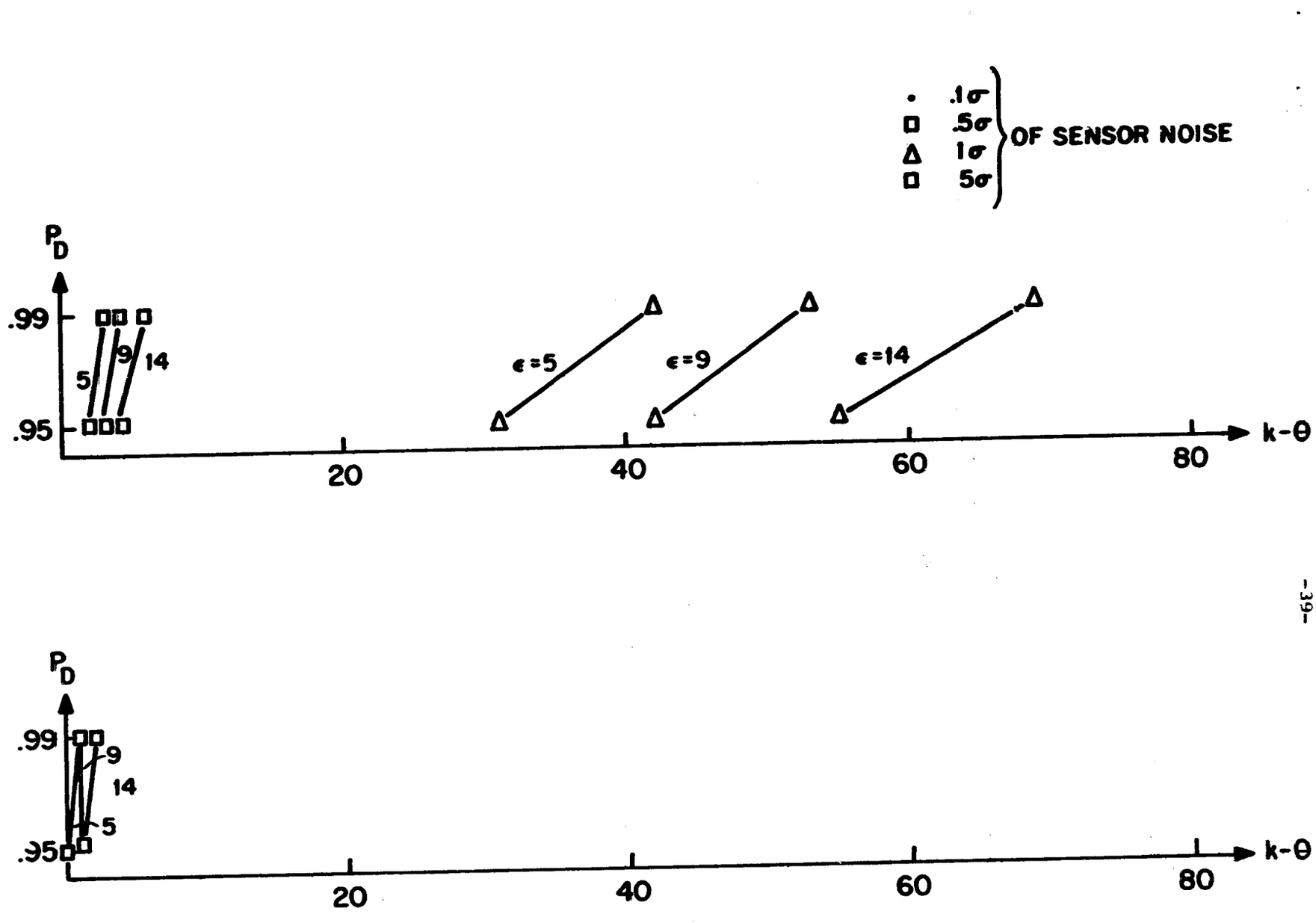


Figure 4.2 Step failure in angle of attack deviation sensor: $V' = [v_1, v_2]$

value (.95, .99, etc.) in a few steps or it never reaches it. This, however, is in agreement with the previous analysis. The effect of a jump failure decays as time progresses thus creating a δ^2 approaching a constant steady state value. For a fixed threshold, this value of δ^2 corresponds to a fixed value of P_D which may be bigger or smaller than the desired P_D. Therefore, jump detector windows should be small since large windows do not necessarily improve detection.

We note that all of these quantities are <u>static</u> quantities -- i.e., P_F and P_D calculated here are simply the probability that $\ell(k; \theta) > \varepsilon$ for fixed $k - \theta$. These numbers should be interpreted as follows:

1) Fix $k - \theta = r_0$. We are looking <u>only</u> for failures at the time $\theta = k - r_0$, and thus at any time k we need only evaluate <u>one</u> likelihood ratio, $\ell(k; k - r_0)$, using the window of residuals $\gamma(k - r_0)$, $\gamma(k - r_0 + 1)$, ..., $\gamma(k)$.

2) The numbers P_D and P_F in the figures are the probabilities that

$$\ell(k; k - r_0) > \varepsilon$$

under the failure (of size ν) and no failure hypotheses, respectively.

Note however that the given window of residuals can be used to calculate other values of $\ell(k; \theta)$ with

$$k - r_0 \leq \theta \leq k$$

One can then consider questions such as the following: suppose we wish to detect failures anywhere in the interval $[k - r_0, k]$; suppose we define the detection rule:

Declare failure in the interval $[k - r_0, k]$ if K of the likelihood ratio $\ell(k; \theta)$, $\theta \in [k - r_0, k]$ exceed a given threshold ε.

By using more of the data over an interval, such as in this detection rule, we would expect better detection performance -- i.e. by taking $K > 1$, we can reduce P_F, since the effect of one bad data point is somewhat alleviated; on the other hand, by looking at more than one of the $\ell(k; \theta)$, we should increase the probability of detecting failures. However, the calculation of P_D's and P_F's for such decision rules, considerations of the values of M, ε, r_0 and the number of $\ell(k; \theta)$ to be evaluated in the interval are difficult since the ℓ's are correlated non-central χ^2 variables. As mentioned earlier, the situation is somewhat better in the SGLR case, and we plan to consider this in the near future.

V. Simulation Results

V.1. Introduction

The results which the simulations present are rich in content. However, there seems to be so much information in them that full appreciation of it all will take some time and much study of the data. Consequently, a simple physical model must be used in this initial stage until we understand better the limitations in detector performance. Added degrees of freedom at this point would increase the difficulty of interpretation significantly.

In the following sections we take a look at these results. As a first attempt at organizing them we will comment on the overall behavior of the detectors in section V.2. Then we narrow our perspectives to try to draw some conclusions. In section V.3 we focus on jump failures, both in the state and sensor equations and in V.4 we do the same for both kinds of step failures. Section V.5 considers the problem of false alarms and the sensitivity of detection to changes in the threshold. In section V.6 we take a look at the elements of $G_i(k;\theta)$, the failure signatures and some interesting differences in detection performance for failures in the pitch rate and angle of attack directions with some physical interpretation are seen in V.7.

It is hoped that the qualitative descriptions and physical interpretations of the detector behavior show why the GLR approach makes sense. All this will provide a useful evaluation of the GLR detectors as well as of our understanding of it to date. These results should suggest the next steps to be taken.

All the simulations were done with the corresponding detector for each type of failure, i.e. for a failure of type i the detector implemented is based on $G_i(k;\theta)$. (A different problem is that of cross-detection: looking for a failure of type i with a detector based on $G_j(k;\theta)$ with $j\neq$). The decision threshold was fixed at the value $\varepsilon=5$ all throughout and window sizes were [M=10, N=0] for jump detectors and [M=30, N=0] for step detectors. All failures occurred at $k=5\overset{\Delta}{=}\theta_T$, and the simulations ran five time units past the moment when θ_T, the true time of failure, left the window. Recall, the window at time k is composed of $\{k-M, k-M+1,\ldots,k-N-1, k-N\}$. Therefore the simulation stopped at $k=M+\theta_T+5=M+10$, or k=20 for jumps and k=40 for steps. The flowgraph in Figure 5.1 illustrates the general make-up of the simulations.

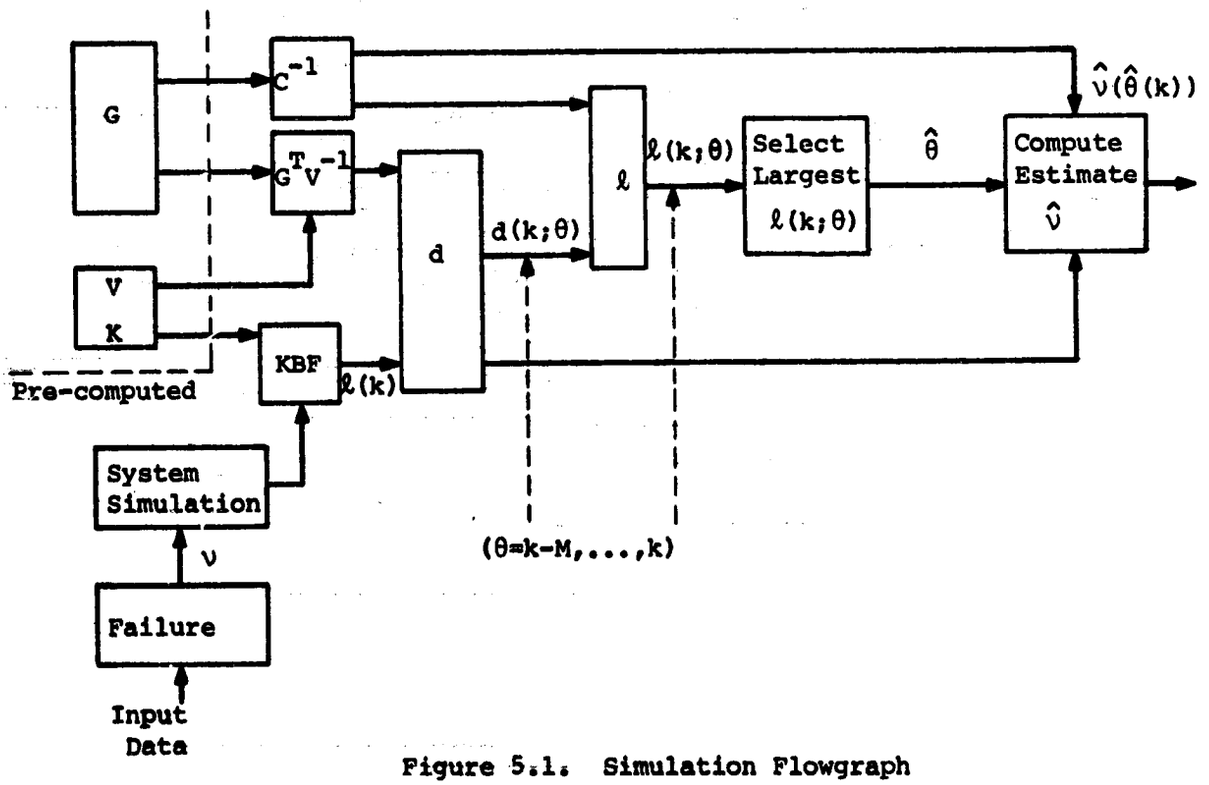


Figure 5.1. Simulation Flowgraph

The system and filter matrices, threshold ε, failure time θ_T, window size M,N and failure are specified at the beginning of a run. Then the detector matrices $G(k;\theta)$, $C(k;\theta)$ are computed. At everytime k during the simulation an observation $Z(k)$ is produced which results in its corresponding filter residual $\gamma(k)$. For every θ in the window specified $d(k;\theta)$ is computed and

$$\hat{\theta} = \arg \max_{\theta} \ell(k;\theta)$$

is selected. Using this estimate of the time of failure, an estimate of the failure is then computed using $d(k;\hat{\theta}(k))$ and $C^{-1}(k;\hat{\theta}(k))$ with the equations presented in section III.5. This procedure is repeated again for k+1 until the final time is reached.

V.2 Description of simulation results

Here we provide a brief qualitative description of the detector performance for the different failures considered. Recall the decision rule at time k,

$$\text{Detection when } \max_{\theta} \ell(k;\theta) \geq \varepsilon = 5.0, \quad \theta \in \{k-M,\ldots,k-N\} \cap \{\theta \mid \theta > 0\}$$

The set of simulations consisted of failures of the four types presented in Section III.6 for the range of values shown in Table 3.1. For each case two runs were made, each with different noise sequences for the process and measurement noises. Although two runs do not provide statistically significant results, they do allow us to avoid some unjustified generalizations

based on a single set of data. Some of the simulation outputs are presented in sections V.3, V.4 and V.5.

We begin with a description of the results for the case of state failures.

Jumps:

1σ .... Small delay of 2-4 time steps ($\frac{1}{16}$ sec - $\frac{1}{8}$ sec.) before detection of pitch rate failures, $(\nu,0)^T$; detection is immediate in the angle of attack direction. The estimation of the value of $\underline{\nu}$ is erratic. When θ_T, the true time of failure, leaves the window, even detection itself degrades as it becomes more sensitive to the noise.

$\geq 5\sigma$ ... Detection is immediate in all cases seen. We will see in section V.5 that this remains true even for threshold at least as high as $\varepsilon=14$. Correct identification of θ_T takes place, especially for the larger failure magnitudes, 10σ and 20σ. The estimates $\hat{\underline{\nu}}$ are less erratic although the best estimate is attained in a few time steps (≤ 5) with no further improvement. Estimates degrade rapidly as soon as θ_T leaves the window.

Steps:

$1/10\sigma$ ... Detection takes place, although it is somewhat erratic: it may be lost for varying lengths of time. While not very accurately or consistently, the fact that a failure has occured can be

ascertained. The simulations indicate some sensitivity of detection to the noise processes in the system: one run showed delays (3 and 13 time steps) in detection while the other one showed no delays. The larger delay is in the α direction, the angle of attack. The estimated time of failure varies and the failure estimates are not close to the true failures.

$1/2\sigma$ .. Detection is fast: largest delay was 5 time steps. Apparent sensitivity to noise in detection because for one run detection was immediate. Also, no significant difference is seen in delays to detection for failures in q and α. The estimate $\hat{\theta}$ of the failure time goes through a small transient and settles near θ_T (± 1 or 2) The failure estimate is slightly more accurate than for $\frac{1}{10}\sigma$ failures but it degrades rapidly as k increases.

1σ .... Detection is very fast: either immediate or with a delay of 1 time step for failures in both q and α. The estimates $\hat{\theta}$ are correct and show improved accuracy in $\hat{\underline{\nu}}$ over $1/2\sigma$ failures, mainly for failures in q, $(\nu,0)^T$. Best estimate $\hat{\underline{\nu}}$, reached after approximately 15-20 time steps with slow degradation thereafter.

5σ ... Detection is excellent: it takes place without any delays for failures in both directions, q and ν. The estimate $\hat{\theta}$ goes to θ_T very quickly and the estimation accuracy in $\hat{\underline{\nu}}$ increased significantly over the previous cases ($1/10\sigma$ to 1σ steps). For

example, the estimate $\hat{\nu}_i$ of the non-zero element of $\underline{\nu}$ ($\nu_i=\nu_1$ for q and $\nu_i=\nu_2$ for α) comes within 5-10% of the true value and $|\hat{\nu}_i| >> |\hat{\nu}_0|$, where $\hat{\nu}_0$ is the estimate of the other element of $\underline{\nu}$ (true value is zero).

Before going on to the sensor failures let us note a few things. First of all, since one time step in this model is 1/32 sec., a delay in detection of 8 time steps, for example, represents in real time a delay of 0.25 sec. Delays must therefore be fairly long and errors in $\hat{\theta}$ must be fairly large in order for them to be significant.

Next a comment on the accuracy of the failure estimates. It can be shown that $C^{-1}(k;\theta)$ has the interpretation of being the covariance matrix of the error in the estimate $\hat{\underline{\nu}}$. In Figures 5.2 and 5.3 we have plotted the elements of $C^{-1}(k;\theta)$ for the cases of state jumps and state steps for our model. Note that for jumps steady state value are reached almost immediately. This means that for this type of failure the estimate obtained initially, after a few time steps, is as good as we can expect to obtain. Alternatively, for state steps we note from Figure 5.3 that if one waits 20 time steps (5/8 sec.) our confidence in the estimate increases considerably.

This is borne out by the simulation results. Figures 5.4 and 5.5 are plots of the estimate in the phase plane for state jumps and steps of magnitude 1σ and 5σ. Note that the estimate in the jump cases results in greater error and does not improve significantly with time. In contrast to

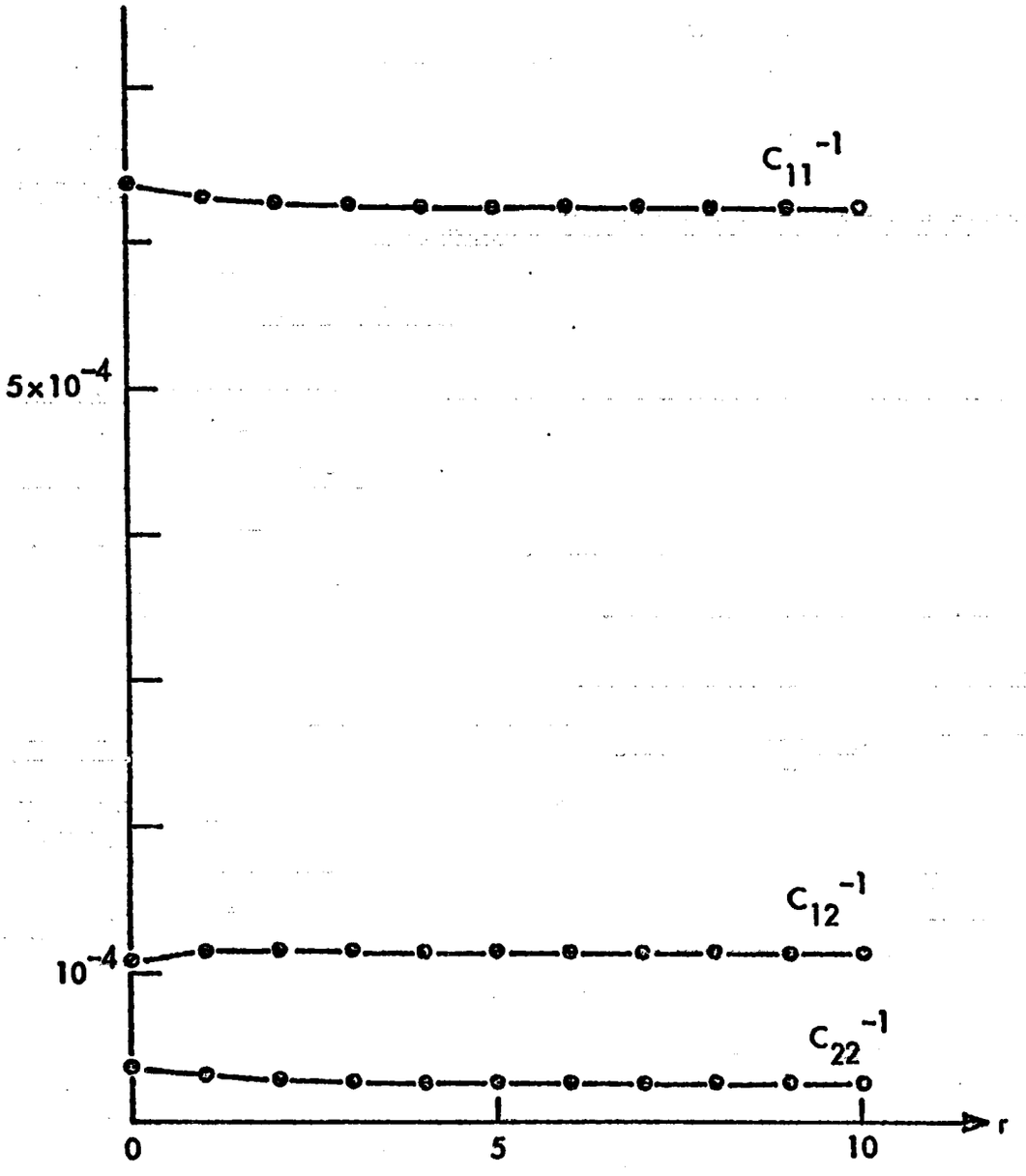


Fig. 5.2 State Jump $C^{-1}(r)$, $r = k - \theta$

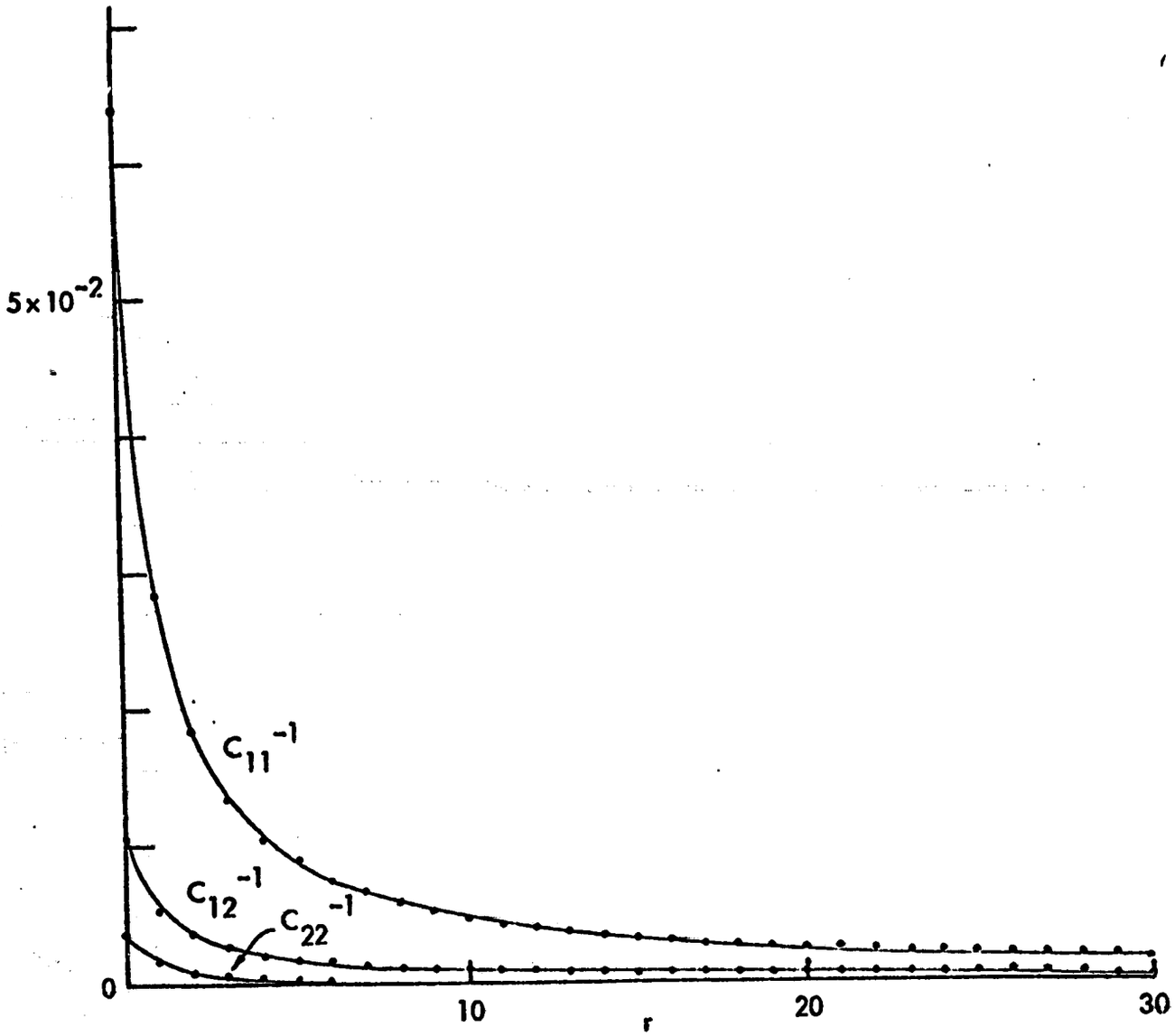


Fig. 5.3 State Step $C^{-1}(r)$, $r = k-\theta$

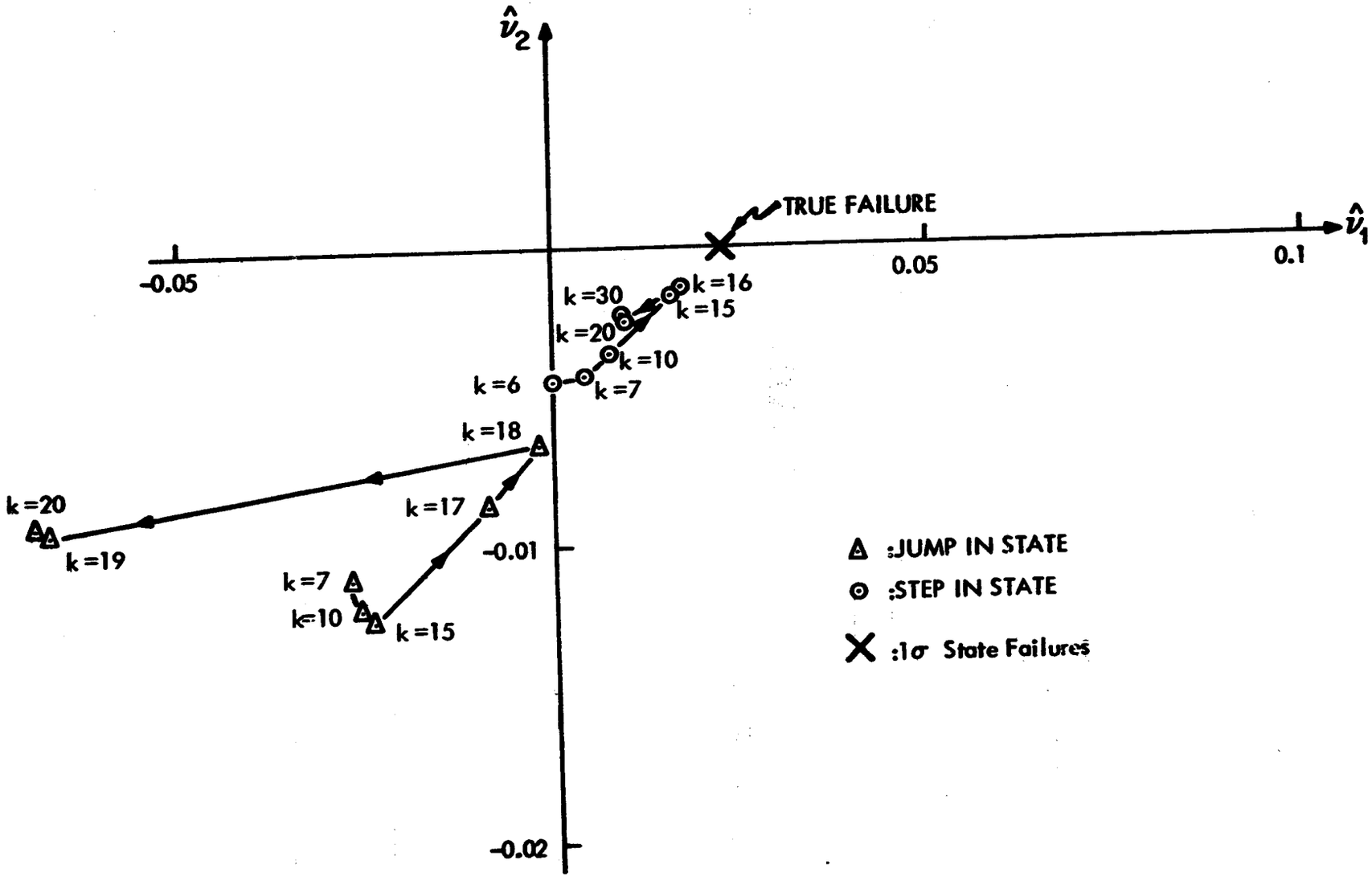


Fig. 5.4 Phase Plane Plot of the Estimate: 1σ State Failures

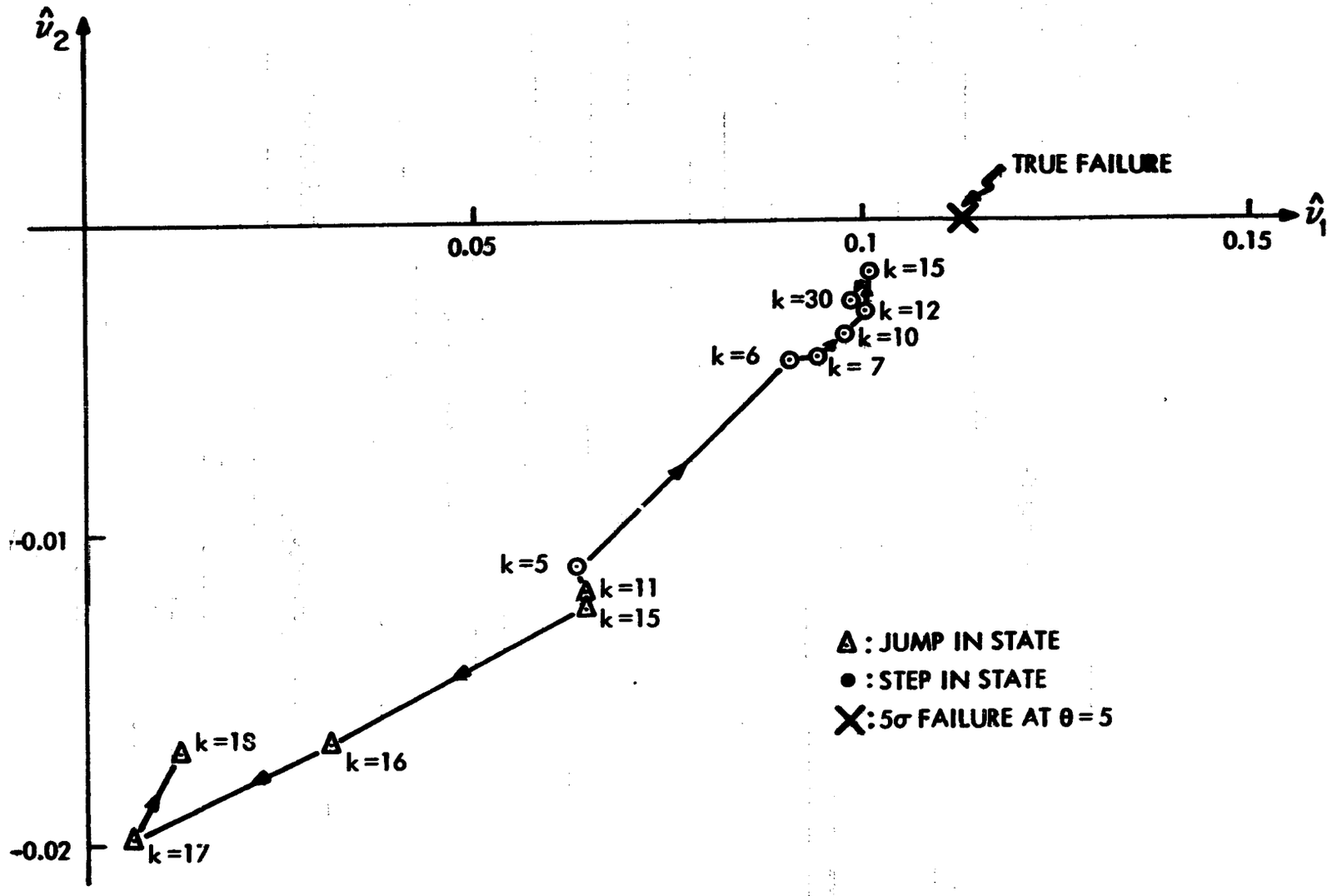


Fig. 5.5 Phase Plane Plot of the Estimate: 5σ State Failures

this we see that for steps the estimates gradually approach the true values. We now take a look at the sensor failures.

Jumps:

$1\sigma'$ ... Detection is immediate and is maintained while θ_T remains inside the window. As soon as θ_T leaves the window, detection becomes erratic. The estimate $\hat{\theta}$ is very sensitive to noise even while θ_T is in the window. Also, the estimate $\underline{\hat{\nu}}$ is not very accurate although there is some improvement while θ_T remains in the window.

$5\sigma'$ ... Detection is quick: immediate in most cases, a delay of 5 time steps showing up in one sample run for a q failure. Otherwise, no significant improvement in performance over the above 1σ case except for slightly more accurate estimation of $\underline{\nu}$.

$\geq 10\sigma'$ ... Detection is excellent: immediate in all runs except for one, with a delay of one time step, in q. In estimation, $\hat{\theta}$ is correct and the failure estimates $\underline{\hat{\nu}}$ are much more accurate especially for angle of attack failures. For example, for a failure in α, $(0,\nu_2)^T$, of magnitude $20\sigma'$ the estimate $\underline{\hat{\nu}}^T = (\hat{\nu}_1,\hat{\nu}_2)^T$ is such that $\hat{\nu}_2$ is within 5% of the true value with $|\hat{\nu}_2| \geq 10|\hat{\nu}_1|$. For comparison, the estimates for a failure in q, $(\nu_1,0)^T$, are such that $\hat{\nu}_1$ is within 25% of the true value and with $|\hat{\nu}_1| \tilde{=} |\hat{\nu}_2|$.

Steps:

$1/10\sigma'$ ... Detection is excellent, it is immediate for failures in both q and α. The estimate $\hat{\theta}$ is very sensitive to noise and the failure estimate $\hat{\nu}$ is very erratic.

$1/2\sigma'$ ... Performance of detectors is similar to the above for $1/10\sigma'$: detection is immediate with $\hat{\theta}$ responding to noise. Slight improvement in estimating $\underline{\nu}$.

$1\sigma'$ ..... Immediate detection. Some improvement over $1/10\sigma'$ and $1/2\sigma'$ failures in the estimation of $\underline{\nu}$.

$5\sigma'$ ..... Generally very good performance: immediate detection except for one run for a failure in q. The estimate $\hat{\theta}$ is very close to θ_T and the estimates of the failures show significant improvement over the above cases.

$\geq 10\sigma'$ ... Very good detection in general. Detection is immediate in all cases except for a run with a delay of one time step for a failure in q. The estimate $\hat{\theta}$ goes quickly to $\theta_T \pm 1$. The failure estimates are relatively accurate: within 5% of true value, although they degrade gradually after θ_T leaves the window. Results are better for the larger failure magnitudes.

Once again let us make some remarks. Figure 5.6 is a plot of $C^{-1}(k;\theta)$ for the case of sensor jumps. We see that our confidence in the estimate, inversely proportional to $C^{-1}(k;\theta)$, does not improve noticeably after the initial time step. In contrast to this, in Figure 5.7 $C^{-1}(k;\theta)$ is plotted

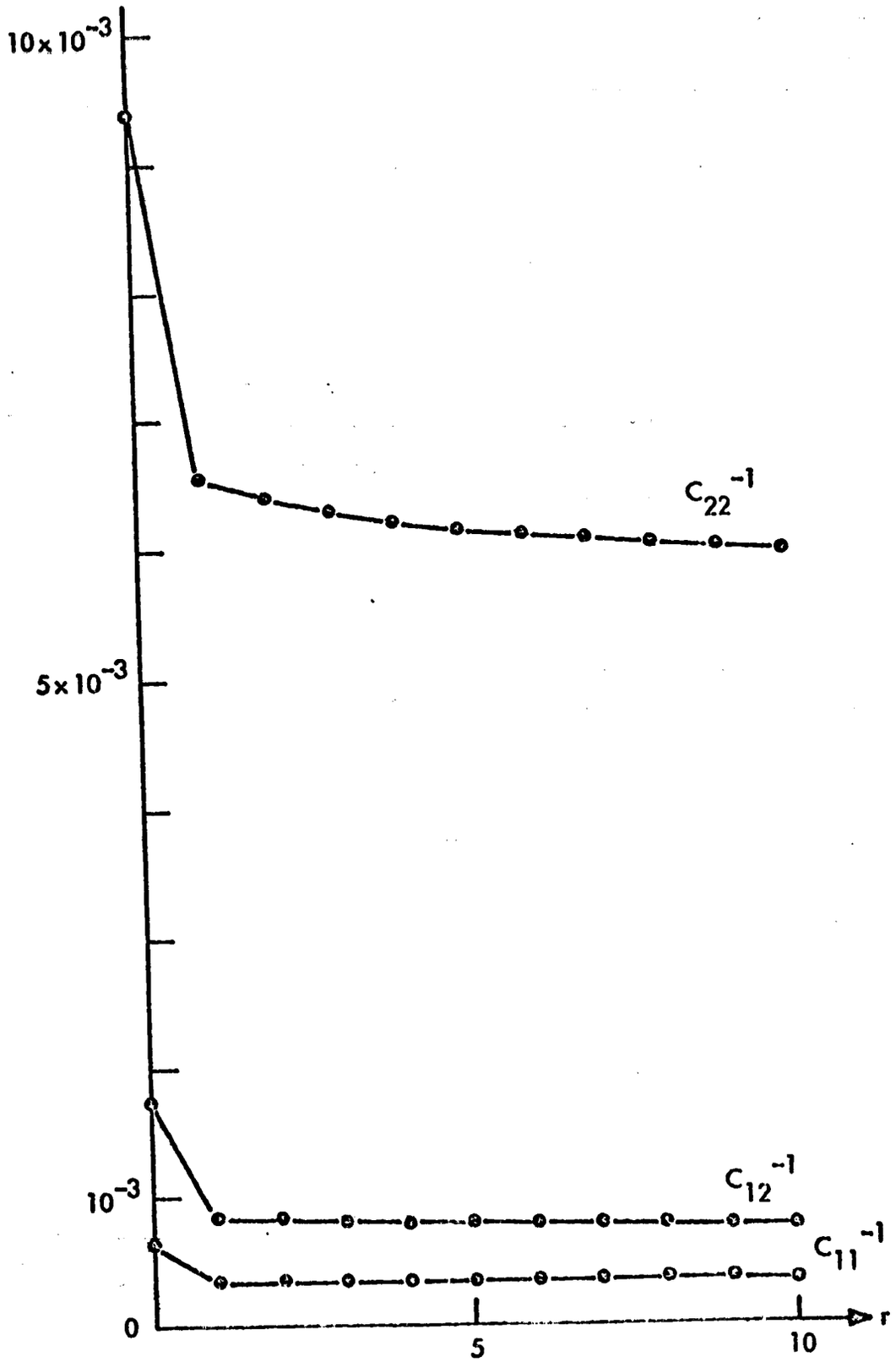


Fig. 5.6 Sensor Jump $C^{-1}(r)$, $r = k - \theta$

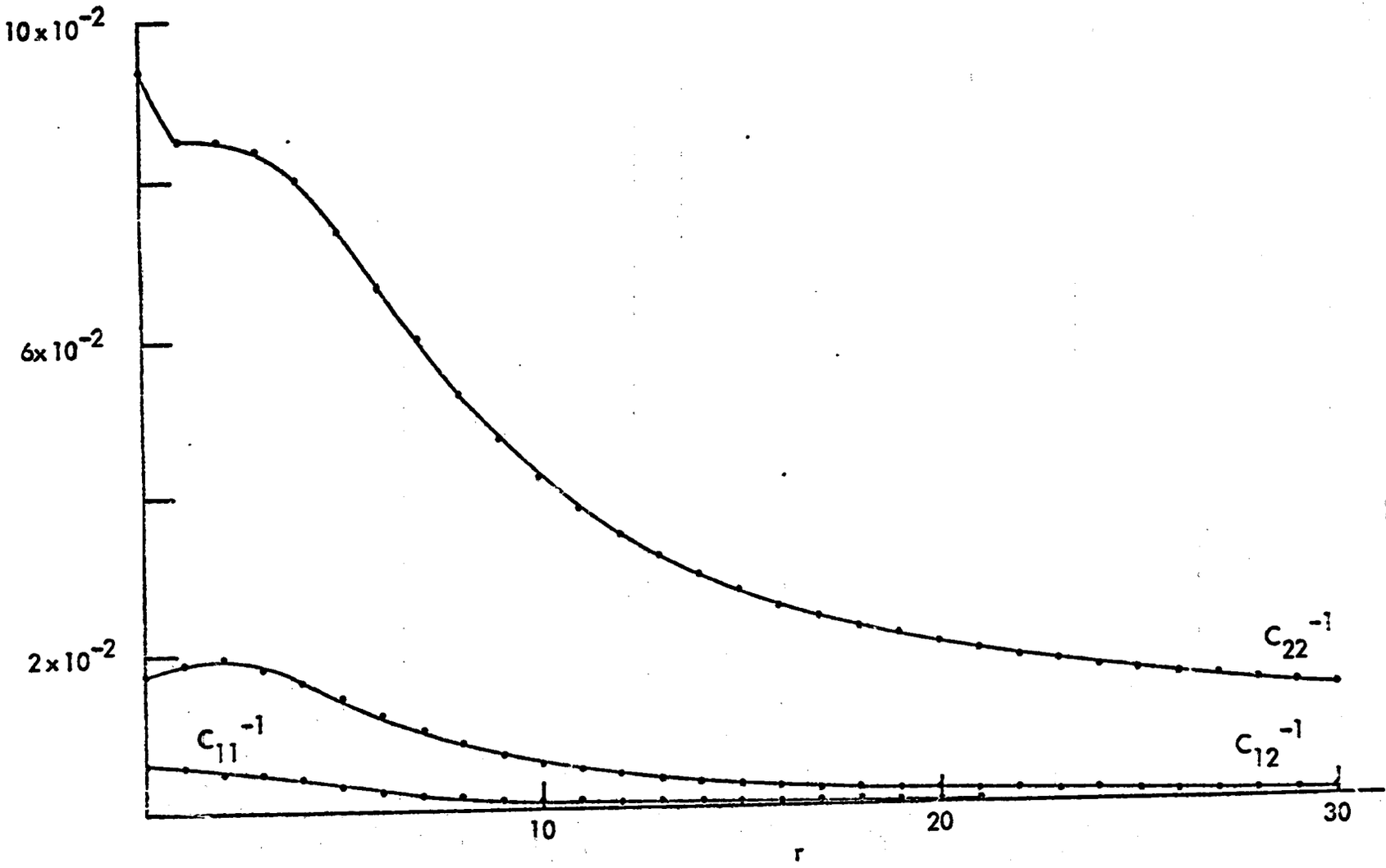


Fig. 5.7 Sensor Step $C^{-1}(r)$, $r = k - \theta$

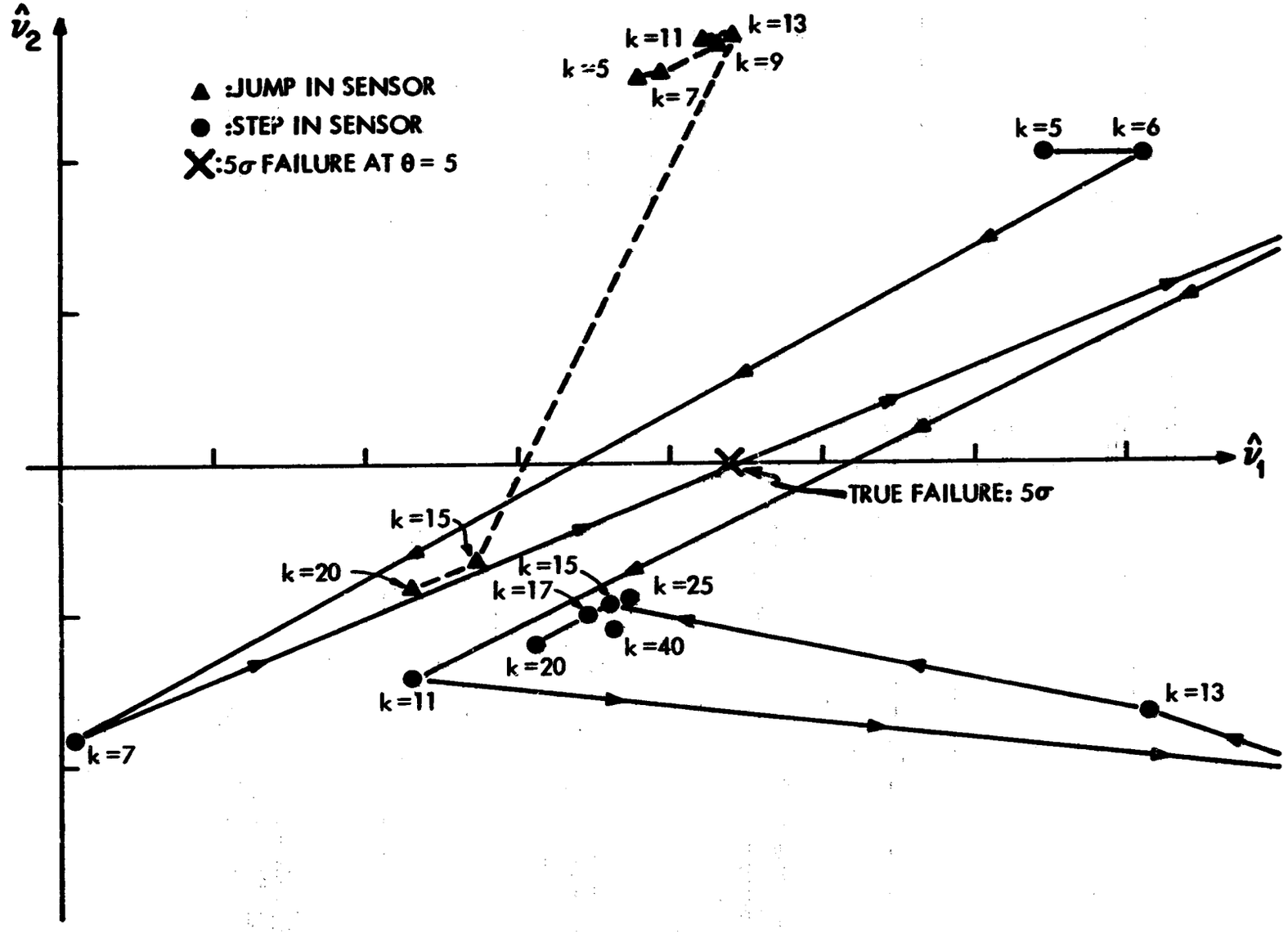


Fig. 5.8 Phase Plane Plot of the Estimate: 5σ Sensor Failures

for sensor steps and in this case as k-θ increases our certainty in the estimate does too. So again for jumps we find that the estimate does not improve significantly over the value obtained after a couple of time steps, we know that we can't do better than that in estimating $\underline{\nu}$. On the other hand, in the case of steps we see that a wait of 20 time stpes decreases $C^{-1}(k;\theta)$, the covariance of the estimation error, significantly. Figure 5.8 contains plots of the estimate of the failure produced by the GLR detectors for a 5σ' failure in the sensor for q, both jump and step. We see that, as mentioned above, the estimate in the jump case improves little with time while that of the step case achieves its best values after 10 time steps ($k \geq 15$).

Before going on to the next section one last thing will be mentioned which provides some background for what follows and yields some insight into the dynamics of the detector. For failures of all kinds considered except for sensor jumps, when θ_T drops out of the window the detector selects θ_T+1 and then θ_T+2, θ_T+3,... as the value of $\hat{\theta}$. By the definition of $\hat{\theta}$, those values of $\hat{\theta}$ correspond to θ with the largest $\ell(k;\theta)$ in the window. We will try to understand this by means of an example.

Let us consider a simplified model of the aircraft dynamics which is valid over short periods of time. We can set the angle of attack α to be the integral of the pitch rate as a first order approximation.

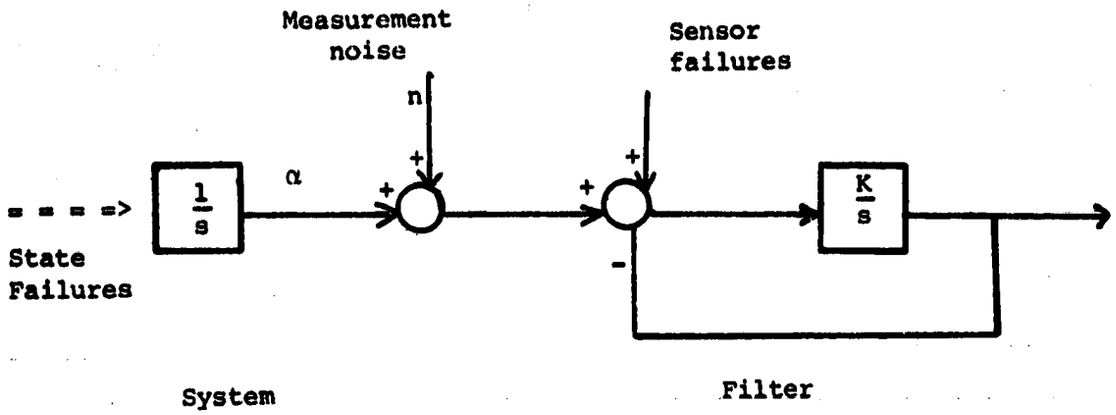


Figure 5.9

A jump in q here looks like a step in α to the filter which can only measure α. This similarity tells us that in the future there might be some difficulty in cross-detection for these two types of failures.

For this system the filter can track a step input (with zero steady-state error) and thus $G(k-\theta)$ goes to zero as $k-\theta$ increases for a jump in q or step in α, which to the filter look the same. However, for a step in q, this does not happen since it leads to a ramp at the input to the filter. For this input the filter has a steady state tracking error and thus $G(k-\theta)$ does not go to zero for this case. Therefore if we do not detect a state jump or a sensor step or jump quickly, it will go undetected. In the case of a step in q, however, it leads to a sensor ramp as input to the filter which leads to a persistent effect on the residuals. This means that one will get more and more information about such a mode as time goes on.

We now have a sensible explanation for the incrementation of $\hat{\theta}$ once θ_T drops out of the window as is the case for state steps, for example. When θ_T, the time at which such persistent and possibly increasing effects began,

is no longer a candidate for $\hat{\theta}$ then $\theta_T+j = k-M$ is seen by the detector as the most likely time of occurrance, where k-M is the earliest time in the window. The detector sees that some excitation persists throughout the entire window. Its best guess for its beginning is the earliest time step it can guess: the first point of the window.

V.3 Jump Failures: State and Sensors

For jump failures we find that detection in general is very good. The presence of a failure is identified almost immediately, with possibly some small delay in some of the failures of smaller magnitudes. When translated to real time these delays are 1/8 second or less.

Figures 5.10 and 5.11 point out a basic difference between detection of state jumps and sensor jumps. The graphs represent the values of $\ell(k;\theta)$ for the θ inside the window (those inside the window such that $\ell(k;\theta)>\varepsilon$ in Figure 5.11). Although they are given for different failure magnitudes, one is for a 10σ state jump in q while the other is for a $20\sigma'$ sensor jump, they show typical $\ell(k;\theta)$ profiles over a window. While sensor jumps result in distinctive spikes in the GLR's for the times of failure, state jumps lead to GLR profiles (as function of θ for a given time k) which are smoothed out in some sense. Thus, in general, detection of state jumps is less noise-sensitive than detection of sensor jumps. The system dynamics, in effect, act like a low-pass filter.

As Figures 5.2 and 5.6 showed, for jump failures the estimate after a few time steps is already in some sort of steady state. Waiting does not

```
RESIDUALS AT TIME    11
   6.0560E-03   -3.9004E-01
GLRS(LG)  :   0.5441E+03

OUTPUT OPTION  :     3

*******************************FAILURE OCCURED AT TIME =     5 = θ̂(k=11)
ESTIMATED FAILURE VECTOR =
   1.7706E-01   -1.1984E-02

 TIME        GLR                     GLR BAR GRAPH
   1      0.1843E+03        XXXXXXXX
   2      0.2243E+03        XXXXXXXXXX
   3      0.3003E+03        XXXXXXXXXXXXX
   4      0.3765E+03        XXXXXXXXXXXXXXXXX
   5      0.5441E+03        XXXXXXXXXXXXXXXXXXXXXXXXX
   6      0.3983E+03        XXXXXXXXXXXXXXXXXX
   7      0.3127E+03        XXXXXXXXXXXXXX
   8      0.2262E+03        XXXXXXXXXX
   9      0.1434E+03        XXXXXX
  10      0.7573E+02        XXX
  11      0.3654E+02        X
```

Figure 5.10 GLR's for State Jump of 10σ in Pitch Rate, $(\nu_1, 0)$.

```
RESIDUALS AT TIME    11
   2.0252E-02    9.8844E-02
GLRS(LG)  :   0.6743E+02

*******************************FAILURE OCCURED AT TIME =  5 = θ̂(k=11)
ESTIMATED FAILURE VECTOR =
   1.2801E-01    5.5544E-03

 TIME        GLR                     GLR BAR GRAPH
   1      0.8376E+01        XXXXXX
   5      0.6743E+02        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   9      0.6197E+01        XXXX
  10      0.7947E+01        XXXXX
```

Figure 5.11 GLR's for Sensor Jump of 20σ' in Pitch Rate, $(\nu_1, 0)$.

provide extra information for an improved estimate. As soon as θ_T leaves the window the estimates deteriorate, more gradually for state than for sensor failures.

V.4 Step Failures: State and Sensors

For the case of step failures both in the state and sensors, detection is excellent with some delay for the smallest failures tried: $1/2\sigma$, $1/2\sigma'$, $1/10\sigma$ and $1/10\sigma'$. The largest delay found was for a $1/10\sigma$ failure in the state for angle of attack in which case it consisted of 13 time steps, or 0.4 sec. This failure represents a step in angle of attack of 2.2603×10^{-5} radians or approximately 0.0013 degrees.

As far as the estimation of the correct time of failure is concerned, given our fixed threshold in the simulations the critical factor is the size of the failures. In general we find that the estimate improves with time and so $\hat{\theta}$ undergoes a kind of transient and then tends to θ_T. The reason for this is that the GLR's for the step cases grow in time, at least while θ_T remains in the window. For very small failures, $1/10\sigma$ and $1/2\sigma$, $\hat{\theta}$ is very sensitive to noise, which is less important for the larger failures. For state steps greater than $1/2\sigma$, sensitivity to noise is greatly reduced while for sensor steps a similar reduction takes place only for failures greater than 1σ. This is not surprising if we recall the discussion at the end of section V.2. We saw that state step failures are in some sense equivalent to ramps in the sensors and will therefore lead to higher GLR values than sensor steps for the same failure size.

As in the case of state jump failures; we again find the same smoothing effect of the system dynamics on $\ell(k;\theta)$ if viewed as a function of θ, that is, on the GLR profile over the times in the window considered at any one instant. However, we find that step failures in the sensors also manifest this same smoothing property for big enough failures ($10\sigma'$,$20\sigma'$), i.e., with high signal-to-noise ratios. Figures 5.12 and 5.13 show the effect of sensor and state step failures in the pitch rate q, for two different times, on the GLR's. The graphs show the GLR profiles over the window for two different times and indicates their characteristic shapes.

V.5 Threshold, False Alarms and Detection

In this section we discuss some results from the simulations which have a bearing on our attempt to understand how the GLR detectors works and to develop some intuition about its behavior. The discussion so far has emphasized the various failures tried and the range of response of the detectors given by the delay times of detection and the estimates of the failure and the time of failure. . We have hinted at the sensitivity of detector performance to the noise in the system, which mainly concerns us for the problem of detecting small failures. This is very close to the problem of false alarms, that is, the possibility of detecting 'something' when in fact no failure has occured.

Detection has been defined in terms of the decision threshold explicitly by the rule:

RESIDUALS AT TIME 13
2.8396E-02 -4.2215E-02
GLRS(LG) : 0.1310E+02

**********FAILURE OCCURED AT TIME = 9 = $\hat{\theta}(k=13)$
ESTIMATED FAILURE VECTOR =
-7.7192E-03 -1.6029E-01

TIME	GLR	GLR BAR GRAPH
1	0.8101E+01	XXXXXXXXXXXXXXXXXXXXXXXXXXX
3	0.7011E+01	XXXXXXXXXXXXXXXXXXXXXXXXX
4	0.9552E+01	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5	0.5235E+01	XXXXXXXXXXXXXXXXXX
6	0.1155E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
7	0.7280E+01	XXXXXXXXXXXXXXXXXXXXXXXXX
8	0.1216E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
9	0.1310E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
10	0.8603E+01	XXXXXXXXXXXXXXXXXXXXXXXXXXX
12	0.7496E+01	XXXXXXXXXXXXXXXXXXXXXXXXXX

RESIDUALS AT TIME 32
-6.0152E-03 -6.6491E-02
GLRS(LG) : 0.2966E+02

**********FAILURE OCCURED AT TIME = 9 = $\hat{\theta}(k=32)$
ESTIMATED FAILURE VECTOR =
7.9026E-03 -8.3964E-02

TIME	GLR	GLR BAR GRAPH
2	0.2214E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3	0.2395E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4	0.2470E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5	0.2237E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
6	0.2639E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
7	0.2509E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8	0.2949E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
9	0.2966E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
10	0.2437E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
11	0.2182E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
12	0.2308E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
13	0.2120E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
14	0.1759E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXX
15	0.1618E+02	XXXXXXXXXXXXXXXXXXXXXXXXXX
16	0.1746E+02	XXXXXXXXXXXXXXXXXXXXXXXXXXX
17	0.1432E+02	XXXXXXXXXXXXXXXXXXXXXXX
18	0.1538E+02	XXXXXXXXXXXXXXXXXXXXXXXX
19	0.1359E+02	XXXXXXXXXXXXXXXXXXXXX
20	0.1466E+02	XXXXXXXXXXXXXXXXXXXXXXX
21	0.1224E+02	XXXXXXXXXXXXXXXXXXX
22	0.1099E+02	XXXXXXXXXXXXXXXXX
23	0.9238E+01	XXXXXXXXXXXXXXX
24	0.8950E+01	XXXXXXXXXXXXXX
25	0.6575E+01	XXXXXXXXXX
26	0.8142E+01	XXXXXXXXXXXXX
27	0.1365E+02	XXXXXXXXXXXXXXXXXXXXXX
28	0.1243E+02	XXXXXXXXXXXXXXXXXXX
29	0.9753E+01	XXXXXXXXXXXXXXX
30	0.8073E+01	XXXXXXXXXXXXX
31	0.1021E+02	XXXXXXXXXXXXXXXX

Figure 5.12 GLR's for Sensor Step of 10σ' in Pitch Rate, $(\nu_1, 0)$, at k=13,32.

```
  3.3679E-02  -2.9502E-01
GIFS(LG) :  0.2272E+03

********************FAILURE OCCURED AT TIME =     5 = θ̂(k=13)
ESTIMATED FAILURE VECTOR =
   1.1709E-02  -2.5939E-03
 TIME       GLR                    GLR BAR GRAPH
    1    0.1964E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    2    0.2067E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    3    0.2178E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    4    0.2237E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    5    0.2272E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    6    0.2231E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    7    0.2119E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    8    0.1935E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    9    0.1697E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   10    0.1438E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   11    0.1188E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXX
   12    0.8473E+02      XXXXXXXXXXXXXXXXXXX
   13    0.3505E+02      XXXXXXX
RESIDUALS AT TIME    32
   5.4310E-02  -3.3539E-01
GIFS(LG) :  0.1734E+04

********************FAILURE OCCURED AT TIME =     6 = θ̂(k=32)
ESTIMATED FAILURE VECTOR =
   9.5330E-03  -2.3100E-03

 TIME       GLR                    GLR BAR GRAPH
    2    0.1714E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    3    0.1729E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    4    0.1734E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    5    0.1732E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    6    0.1719E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    7    0.1696E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    8    0.1665E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    9    0.1627E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   10    0.1585E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   11    0.1540E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   12    0.1488E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   13    0.1431E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   14    0.1375E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   15    0.1313E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   16    0.1250E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   17    0.1184E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   18    0.1116E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   19    0.1044E+04      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   20    0.9725E+03      XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   21    0.8950E+03      XXXXXXXXXXXXXXXXXXXXXXXXXX
   22    0.8114E+03      XXXXXXXXXXXXXXXXXXXXXXXX
   23    0.7327E+03      XXXXXXXXXXXXXXXXXXXXXX
   24    0.6574E+03      XXXXXXXXXXXXXXXXXXX
   25    0.5780E+03      XXXXXXXXXXXXXXXXX
   26    0.5035E+03      XXXXXXXXXXXXXXX
   27    0.4297E+03      XXXXXXXXXXXXX
   28    0.3585E+03      XXXXXXXXXXX
   29    0.2833E+03      XXXXXXXXX
   30    0.2084E+03      XXXXXX
   31    0.1275E+03      XXX
   32    0.5647E+02      X
```


Figure 5.13 GLR's for state step of 1σ in Pitch Rate, $(\nu_1,0)$, at k=13,32

Detection when $\ell(k;\theta) \geq \varepsilon$, for $k-M \leq \theta \leq k-N$.

One would therefore want to take a look at the changes in detection performance and on the rate of false alarms for different thresholds.

The rate of false alarms that results from a specific implementation of the detector is an important measure of their performance. This is true because false alarm rate is one of the parameters that defines the limits of acceptability in a given application. In the context of self-reorganizing systems, which respond to failures by internally altering the control system logic to maintain given performance indices within specified bounds, a high rate of false alarm would lead to excessive and unneccesary changes. Such reasoning justifies our effort and time spent trying to analyze and then verify the false alarm probabilities for the various detectors.

In Section II the false alarm probability P_F is defined in a way that reduces to

$$P_F = \text{Prob}(\ell(k;\theta) > \varepsilon \,|\, H_0)$$

Some runs were made with no failures to see the rate of false alarms we would get and compare them to the computed values. Table 5.1 summarizes some aspects of the results. The precomputed false alarm probabilities are given for comparison. The simulations were run for the threshold value $\varepsilon=5$ and the numbers for the other values of ε were easily extracted from them. All quantities are based on the average over two sample runs. ND is the total number of times of detection, i.e., the number of times k for which some $\ell(k;\theta)$ exceeded the value of the threshold. NDD, however, is the more

realistic measure of false alarm rates. It is the number of distinct detections: detections declared as different occurrences, i.e., with different $\hat{\theta}$ as time of failure, for example, three detections in a row, at k, k+1, k+2, declaring a failure at a particular $\hat{\theta}_1$ count only as one for NDD but as three for ND. The quantity NTS gives the total number of time steps that the simulations lasted, 20 for jumps and 40 for steps.

The reason for looking at NDD is related to the concept of false alarms implied by the definition of P_F. Since P_F is based on H_0, the no-failure hypothesis, and large values of the noise may be undistinguishable from jump failures, a true test for false alarms should not allow large noise spikes in the 'recent' past. A more accurate experiment to verify P_F would require us to re-initialize the detectors every time there is a false alarm.

Threshold	Computed P_F	State Jump		Sensor Jump		State Step		Sensor Step	
		ND/NTS	NDD/NTS	ND/NTS	NDD/NTS	ND/NTS	NDD/NTS	ND/NTS	NDD/NTS
ε=5	0.082085	0.575	0.125	0.700	0.100	0.475	0.150	0.7625	0.150
ε=7	0.030197	0.175	0.100	0.575	0.075	0.3375	0.0875	0.300	0.125
ε=10	0.006738	0.0	0.0	0.300	0.050	0.2625	0.025	0.1125	0.0625
ε=14	0.000912	0.0	0.0	0.175	0.025	0.1625	0.0125	0.0	0.0

Table 5.1 False Alarm Rates for Different Thresholds

As expected, the measure of false alarms decreases for an increased threshold although there is a discrepancy with the precomputed P_F's. One must keep in mind, however, the approximate nature of the counts in Table 5.1 and the limited amount of data from which they are computed. It is expected that for a much larger data base leading to more statistically significant results, the number NDD/NTS would approach the computed P_F's.

Figure 5.14 is a plot of $\max_{\theta} \ell(k;\theta)$, the largest GLR at time k for θ in the window, for the state detectors with no failures. By considering various thresholds we can see how the number of false alarms would change. Although these curves are qualitatively representative of detector performance under the condition of no failure one must keep in mind that this was for one particular run, and therefore true for a particular noise sequence. The actual values plotted would be different for another sample run.

In general, raising the threshold to a value of $\varepsilon=7$ eliminates a significant number of false alarms. Most of them are removed when $\varepsilon=10$. However, the threshold is limited by the specified probability of correct detection.

In raising the threshold we make detection of small failures more difficult and we reduce the correct detection probability. Large failures ($>>1\sigma$ or $1\sigma'$) are not affected because the GLR's reach very high values almost immediately. As failures of smaller magnitudes are tried however, a raised threshold results in delays before detection and in the possibility of missing the failure altogether if it is very small. This is especially

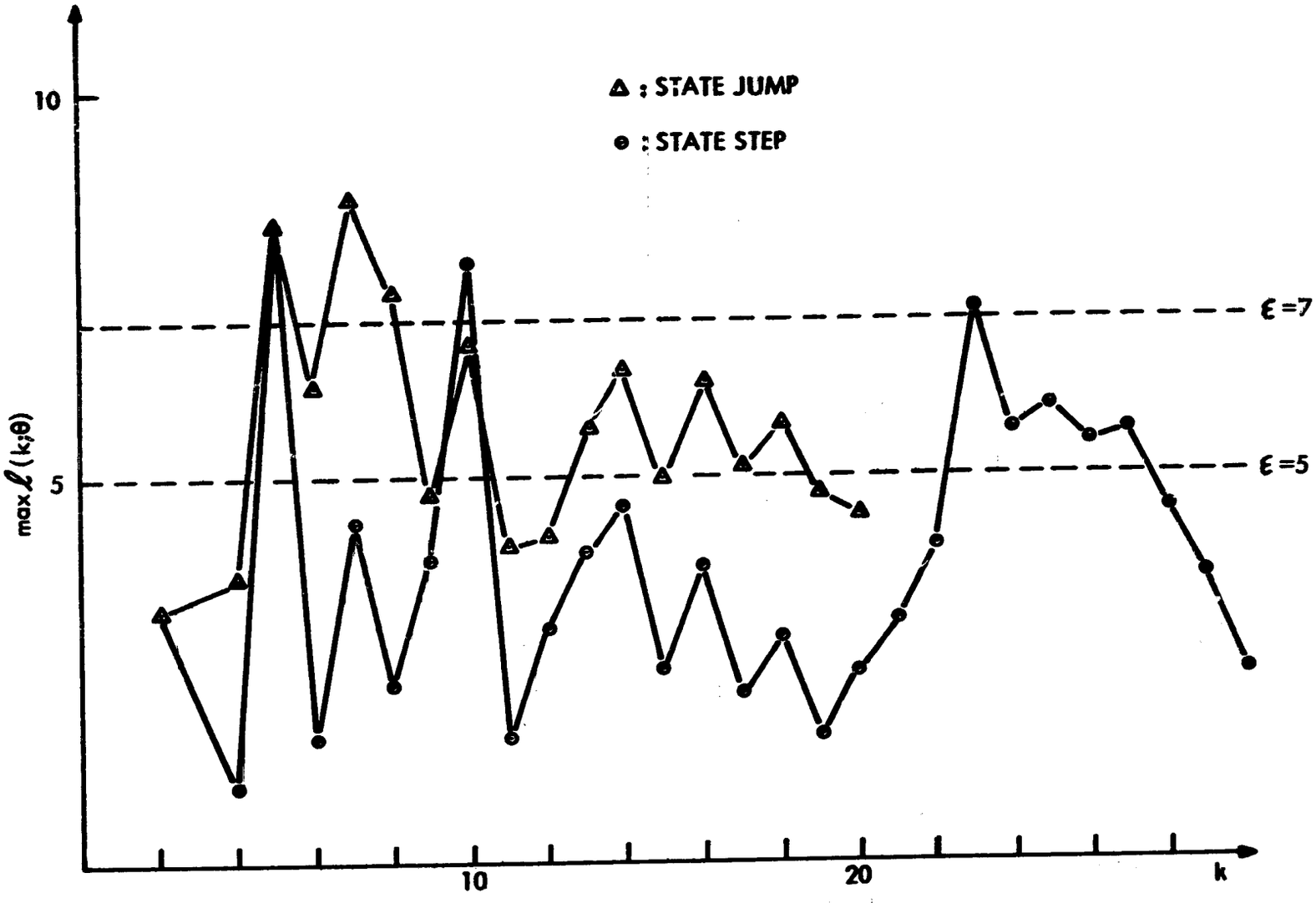


Fig. 5.14 False Alarms: Plot of $\max_{\theta} \ell(k;\theta)$ vs. k for $\nu = 0$

so for small jump failures in the sensors, since the GLR's reach the maximum value quickly and decrease afterwards.

Thus we see that this is a very practical question of performance acceptability. It is possible that for some applications we know a priori that no small failures occur, or perhaps we are not concerned about them, then a higher threshold might be advisable. In conclusion, this is a question to be answered by the problem at hand and by the minimum standards of performance that are specified.

From the simulation data it is possible to infer the detection behavior for higher thresholds. We find the following:

Jump Failures

For jump failures in the state of magnitude greater than or equal to 5σ, detection is virtually unaltered when the threshold is raised from its original value, $c=5.0$, except for the decrease in false alarms. In the case of 1σ state failures however, a number of originally correct detections are eliminated along with the false alarms. This is mostly for $k-\theta_T$ small so in effect we introduce a delay to the time of detection by raising the threshold.

This is summarized in Table 5.2 where the delays in detection of failures are shown for state and sensor jumps of different magnitudes and for additional values of the threshold. For a given failure and threshold, the two entries are the delays after θ_T until the first detection for two sample runs differing only in the noise sequence followed. An entry of ∞ simply means that for those values there was no detection at the end of the run, at $k=20$

for jump failures and k=40 for step failures. It is probable that for a longer waiting time detection would take place. Note the difference between the 1σ state failures and those of larger magnitudes.

Sensor failure detection shows similar effects. The difference in detection between the two kinds of failure lies in the range of failure magnitudes below which even correct detection is affected (besides false alarm rates). For sensor jumps this degradation in performance is seen even for 5σ' failures, although slightly less pronounced for this value. Once again this is seen in Table 5.2 where the same information is shown as for state jumps. Also note that detection of sensor jumps in α is less sensitive to changes in threshold than similar failures in q. This is partly due to the fact that the measurement of α has a higher signal-to-noise ratio than the measurement of q.

Furthermore, we find that for sensor jump the number of false alarms, which generally follow when θ_T drops out of the window, is significantly reduced for a raised threshold. Almost all such false alarms are eliminated for ε=10.

Figure 5.15 shows the values of $\max_{\theta} \ell(k;\theta) = \ell(k;\hat{\theta}(k))$ as it changes with k for 5σ state jumps and 5σ' sensor jumps. The time of failure and that for which θ_T drops out of the window are indicated. Notice the drop in the value of $\ell(k;\hat{\theta}(k))$ for jumps, especially in the sensors, when θ_T

STATE JUMP: $(\nu,0)$					SENSOR JUMP: $(\nu,0)$				
ν	ε=5	ε=7	ε=10	ε=14	ν	ε=5	ε=7	ε=10	ε=14
1σ	2,4	3,4	3,5	7,9	$1\sigma'$	0,0	0,∞	0,∞	4,∞
5σ	0,0	0,0	0,0	0,1	$5\sigma'$	0,5	0,∞	1,∞	∞,∞
10σ	0,0	0,0	0,0	0,0	$10\sigma'$	0,1	0,6	0,∞	0,∞
20σ	0,0	0,0	0,0	0,0	$20\sigma'$	0,0	0,0	0,0	0,0
STATE JUMP: $(0\ \nu_2)$					SENSOR JUMP: $(0,\nu_2)$				
1σ	0,0	0,0	15,0	∞,2	$1\sigma'$	0,0	0,0	0,0	7,∞
5σ	0,0	0,0	0,0	0,0	$5\sigma'$	0,0	0,0	0,0	0,0
10σ	0,0	0,0	0,0	0,0	$10\sigma'$	0,0	0,0	0,0	0,0
20σ	0,0	0,0	0,0	0,0	$20\sigma'$	0,0	0,0	0,0	0,0

Table 5.2 Delays in Detection for Different Thresholds: Jump Failures Measured in time steps from θ_T.

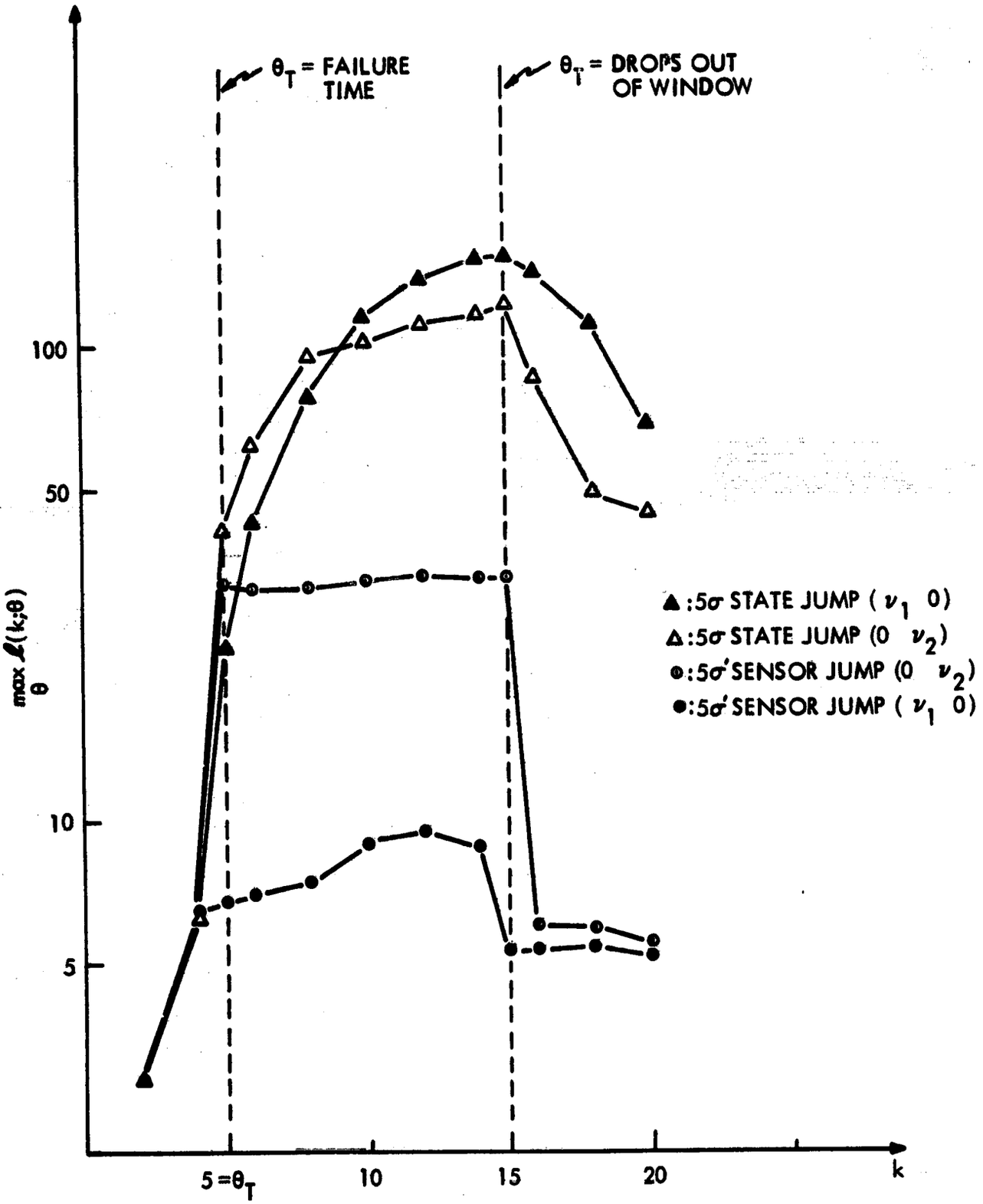


Fig. 5.15 Evolution of $\max_\theta \ell(k;\theta)$ for Jump Failures, Average over Two Runs

drops from the window. As we have seen before, in these cases the measurable effect of the failure on the residuals goes away quickly. Therefore, if we think of the GLR detector as, in a sense, matching the failure signature characteristic of a particular type of failure (as given by $\underline{G}(k;\theta)$), then one would expect that very quickly after θ_T drops from the window $\ell(k;\theta(k))$ would decrease. This happens because there is very little correlation with the information corresponding to θ_T+j, where j is small.

Step Failures

For step failures, once again, sensor failure detection is more sensitive to changes in the threshold than state failures. Whereas for state steps of size $1/2\sigma$ or greater only a small delay to the time of detection is introduced, for sensor steps greater degradation in detection is seen even for $1\sigma'$ steps. A threshold of $\varepsilon=14$, for example, for the given window size used (M=30, N=0) makes $1/10\sigma'$ sensor steps very hard to detect. Notice that if we also increase our window size enough we regain detection, although with a delay which might not be acceptable depending on the application. This is due to the fact that while θ_T remains in the window $\ell(k;\hat{\theta}(k))$ is non-decreasing for the case of sensor steps.

Figure 5.16 is a graph of $\ell(k;\hat{\theta}(k))$ for $1/2\sigma$ state steps and $5\sigma'$ sensor steps as it evolves in time. We can see the generally increasing nature of the GLR while θ_T remains in the window for step failures, as mentioned above. Also note the rapid increase in the GLR soon after θ_T. For higher failure magnitudes, the effect of an increased threshold reduces to the elimination

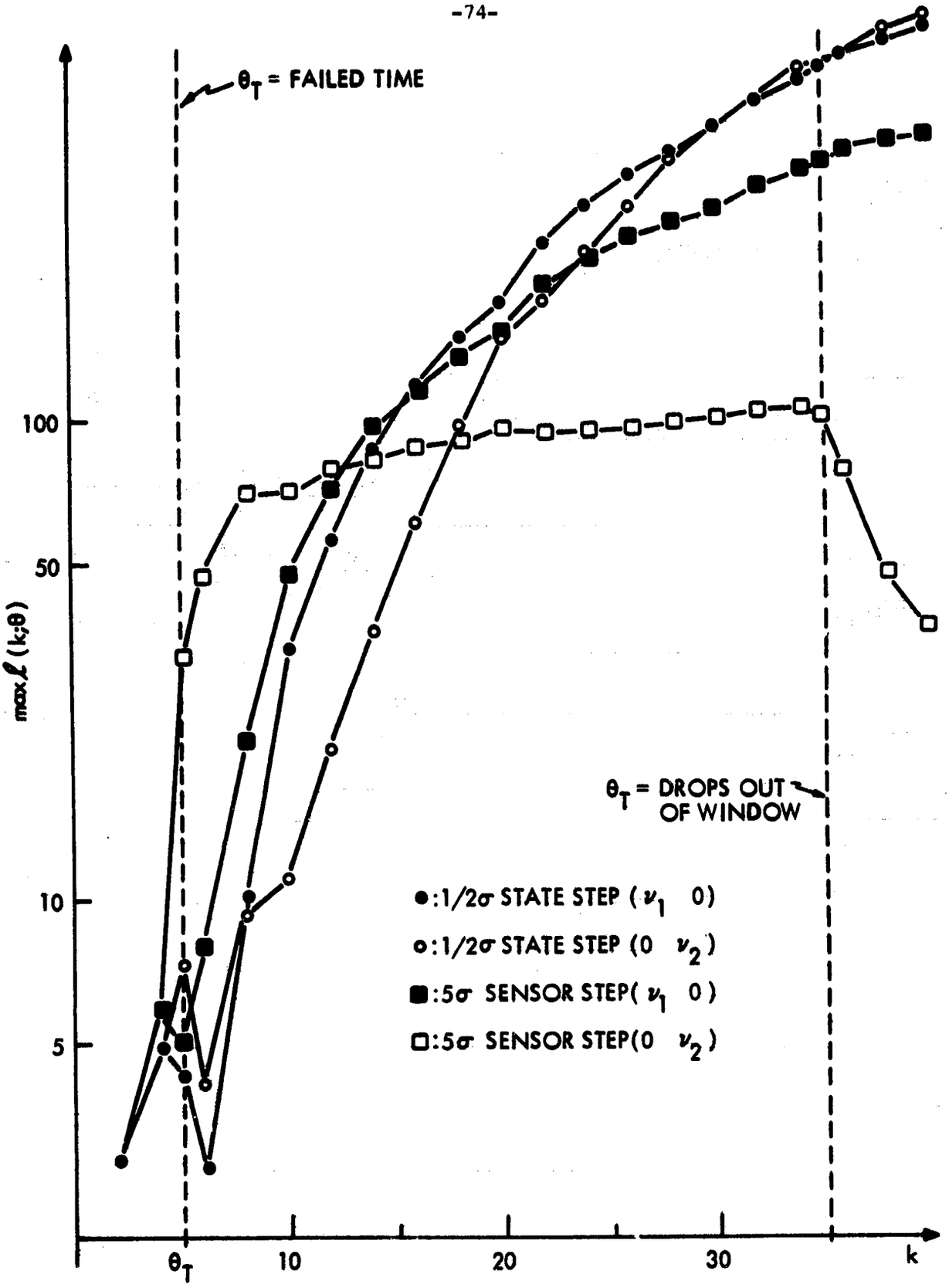


Fig. 5.16 Evolution of $\max_{\theta} \ell(k;\theta)$ for Step Failures, Average over Two Runs

of false alarms and perhaps the introduction of a small delay in some cases, of the order of a couple of time steps. The figure illustrates another fact: the higher values of the GLR for state steps, even for smaller failures than sensor steps. It is a state step which leads to the most persistent effects and this is manifested in these graphs of the GLRS's.

In Table 5.3 we have the delays to detection for step failures and various thresholds just as in the case of jumps. In the case of state steps greater than $1/10\sigma$ detection does not degrade significantly with the increased thresholds. In the case of sensor steps this is true only for small increases in the threshold as large delays may be introduced for somewhat larger thresholds. In general it is true that failures of magnitude 5σ and $5\sigma'$ or greater are detected very rapidly and are not very sensitive to threshold changes of moderate size.

Concluding, higher thresholds look promising if one is interested in moderate sized or large failures. At the price of a possible small delay before correctly detecting the failure, one gains a considerable reduction in the false alarms. If, on the other hand, one is interested in small failures ($\leq 1\sigma$ for state jumps, $\leq 5\sigma'$ for sensor jumps, $\leq 1/2\sigma$ for state steps and $\leq 1\sigma'$ for sensor steps), then the threshold selection must be carefully made. The trade-off between acceptable detection and false alarms is much more sensitive to small changes in the threshold for this range of failures. The study of the various detection probabilities takes on special significance in making such decisions.

STATE STEP: (ν,0)					SENSOR STEP: (ν,0)				
ν	ε=5	ε=7	ε=10	ε=14	ν	ε=5	ε=7	ε=10	ε=14
1/10σ	3,0	4,0	7,5	17,13	1/10σ'	0,0	3,0	4,∞	∞,∞
1/2σ	2,0	3,4	3,4	3,4	1/2σ'	0,0	2,0	4,9	∞,26
1σ	1,1	2,1	2,3	2,3	1σ'	0,0	2,5	3,5	7,9
5σ	0,0	0,0	0,0	0,1	5σ'	0,1	1,3	1,3	2,4
10σ	-	-	-	-	10σ'	1,0	0,1	0,1	0,1
20σ	-	-	-	-	20σ'	0,0	0,0	0,0	0,0
STATE STEP: (0 ν_2)					SENSOR STEP: (0 ν_2)				
1/10σ	13,0	14,0	14,22	14,33	1/10σ'	0,0	3,0	15,∞	27,∞
1/2σ	5,0	5,0	6,0	6,2	1/2σ'	0,0	0,0	15,0	27,∞
1σ	1,0	2,0	2,0	3,2	1σ'	0,0	0,0	15,0	27,2
5σ	0,0	0,0	0,0	0,0	5σ'	0,0	0,0	0,0	0,0
10σ	-	-	-	-	10σ'	0,0	0,0	0,0	0,0
20σ	-	-	-	-	20σ'	0,0	0,0	0,0	0,0

Table 5.3 Delays is Detection for Different Thresholds: Step failures. Measured in time steps from θ_T.

One way to resolve this problem in the case of step failures is to use a longer window, thus allowing the non-centrality parameter, δ^2, to increase and so achieving a higher probability of detection. Another possibility which one might want to look at is the concept of a variable threshold to be used for small step failures. Because of the growing GLR's, after a failure has been detected with a relatively low threshold ε, one could raise the threshold to a value where only the highest GLR's would be accepted. This way the failure effects in some sense are isolated and tracked. This is a question for future consideration.

V.6 $G(k;\theta)$; The Failure Signatures

In this section we will take a closer look at the $G_i(k;\theta)$, the failure signature matrices, which are at the center of all the computations which take place in the detector equations. If we recall, $G_i(k;\theta)$ propagates the effect of a failure at time θ to the residuals of the Kalman filter at time k.

We saw in section 2.2 that because we are working with linear systems, the residuals may be decomposed into two components

$$\underline{\gamma}(k) = \tilde{\underline{\gamma}}(k) + G(k;\theta)\underline{\nu}$$

where $\tilde{\underline{\gamma}}(k)$ is the residual which would be present in the absence of failures and $\hat{\underline{\gamma}}(k) = G(k;\theta)\underline{\nu}$ is the contribution to the residuals from the failure $\underline{\nu}$. The log-likelihood ratio $\ell(k;\theta)$ was seen to be a quadratic in the output of the matched filters, $\underline{d}(k;\theta)$, which in turn are weighted sums of the residuals. Both quantities depend directly on $G(k;\theta)$:

$$\underline{d}(k;\theta) = \sum_{j=\theta}^{k} \underline{G}^T(j;\theta)\underline{V}^{-1}(j)\underline{\gamma}(j)$$

$$\underline{C}(k;\theta) = \sum_{j=\theta}^{k} \underline{G}^T(j;\theta)\underline{V}^{-1}(j)\underline{G}(j;\theta)$$

$$\ell(k;\theta) = \underline{d}^T(k;\theta)\underline{C}^{-1}(k;\theta)\underline{d}(k;\theta)$$

The $G(k;\theta)$ are precomputable and in Figures 5.17 to 5.20 we have plotted the elements of this matrix as functions of $(k-\theta)$ (in our case $G(k;\theta) = G(k-\theta)$ because the system is time-invariant) for the times corresponding to the window lengths implemented. Notice that in the case of state and sensor jumps, Figures 5.17 and 5.18, the elements of

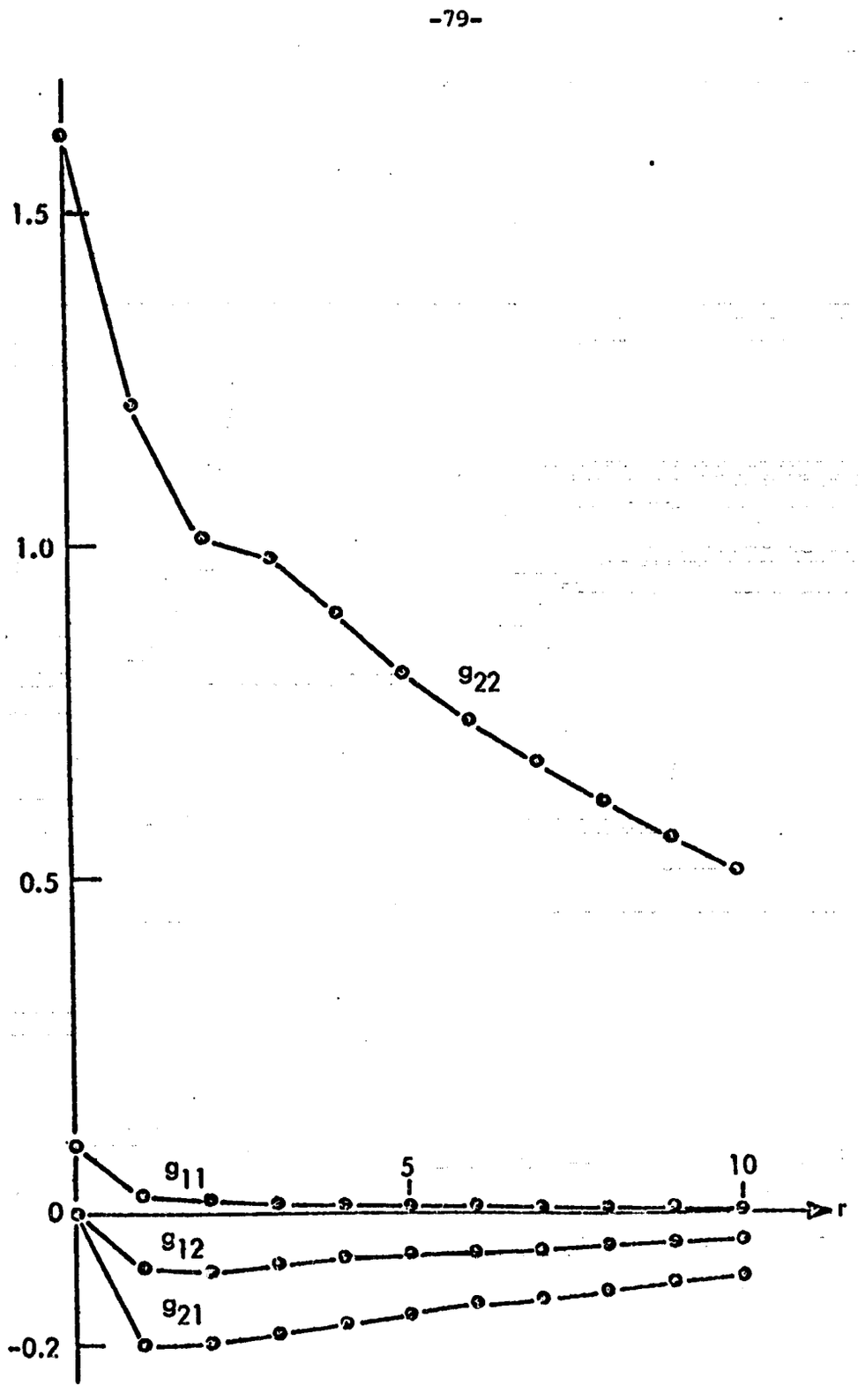


Fig. 5.17 State Jump G(r), r = k -θ

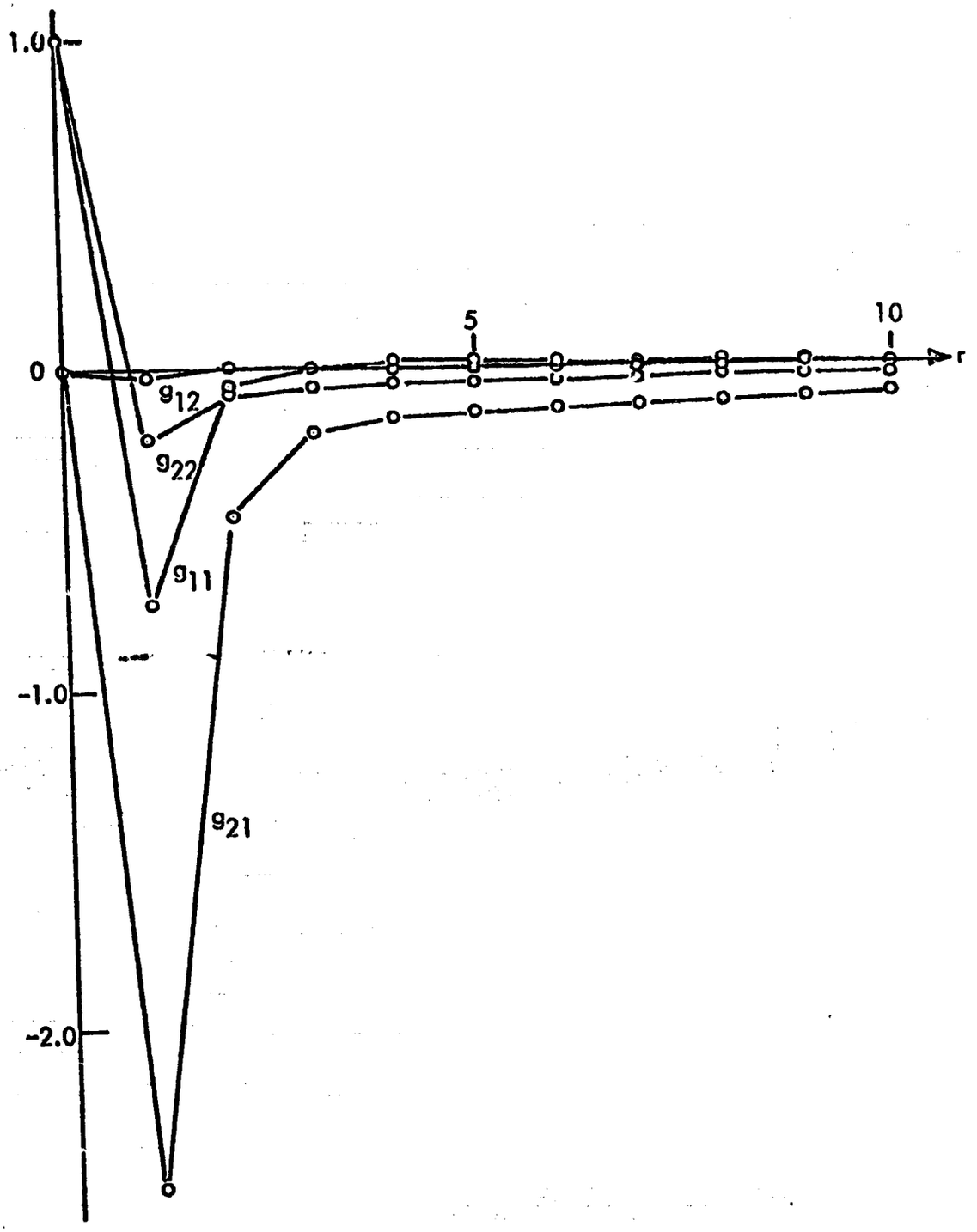


Fig. 5.18 Sensor Jump G(r), r = k -θ

$G(k-\theta)$ die out as $k-\theta$ increases. This is what one would expect qualitatively for a stable system since the effects of an impulse are short-lived. Also note that for sensor jumps the $G_{ij}(k-\theta)$ die out much faster. Recall that the sensor failures pass through one less integration (only the Kalman filter) than state failures (system dynamics and Kalman filter) before reaching the residuals as seen at the end of section V.2. Also the system eigenvalues are near one, the stability boundary for discrete-time systems. It is not surprising then that the effects of state failures on the residuals of the filter in this case persist for a longer time than for sensor failures.

Similarly, in the case of state and sensor step failures shown in Figures 5.19 and 5.20 we see a correspondence to the previously explained fact that their effects are more persistent due to their sustained presence.

Let us make some simple observations about the propagation of the failures to the residuals of the filter. Consider a failure $\underline{\nu}$ of a given type and suppose we have computed the corresponding $G(k;\theta)$ for the appropriate window size. The component $\hat{\underline{\gamma}}(k)$ of the residual due to the presence of the failure is given by $\hat{\underline{\gamma}}(k) = G(k-\theta)\underline{\nu}$ or, in our two-dimensional case,

$$\begin{bmatrix} \hat{\gamma}_1(k) \\ \hat{\gamma}_2(k) \end{bmatrix} = \begin{bmatrix} G_{11}(k-\theta) & G_{12}(k-\theta) \\ G_{21}(k-\theta) & G_{22}(k-\theta) \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}$$

We see that G_{i1} and G_{i2} give the effect of the failure on the i^{th} component of the residual. Alternatively, G_{1j} and G_{2j} give the effect of the j^{th} failure component on the different elements of the vector of residuals.

In our case, with failures in orthogonal directions we get for the case of pitch rate failures

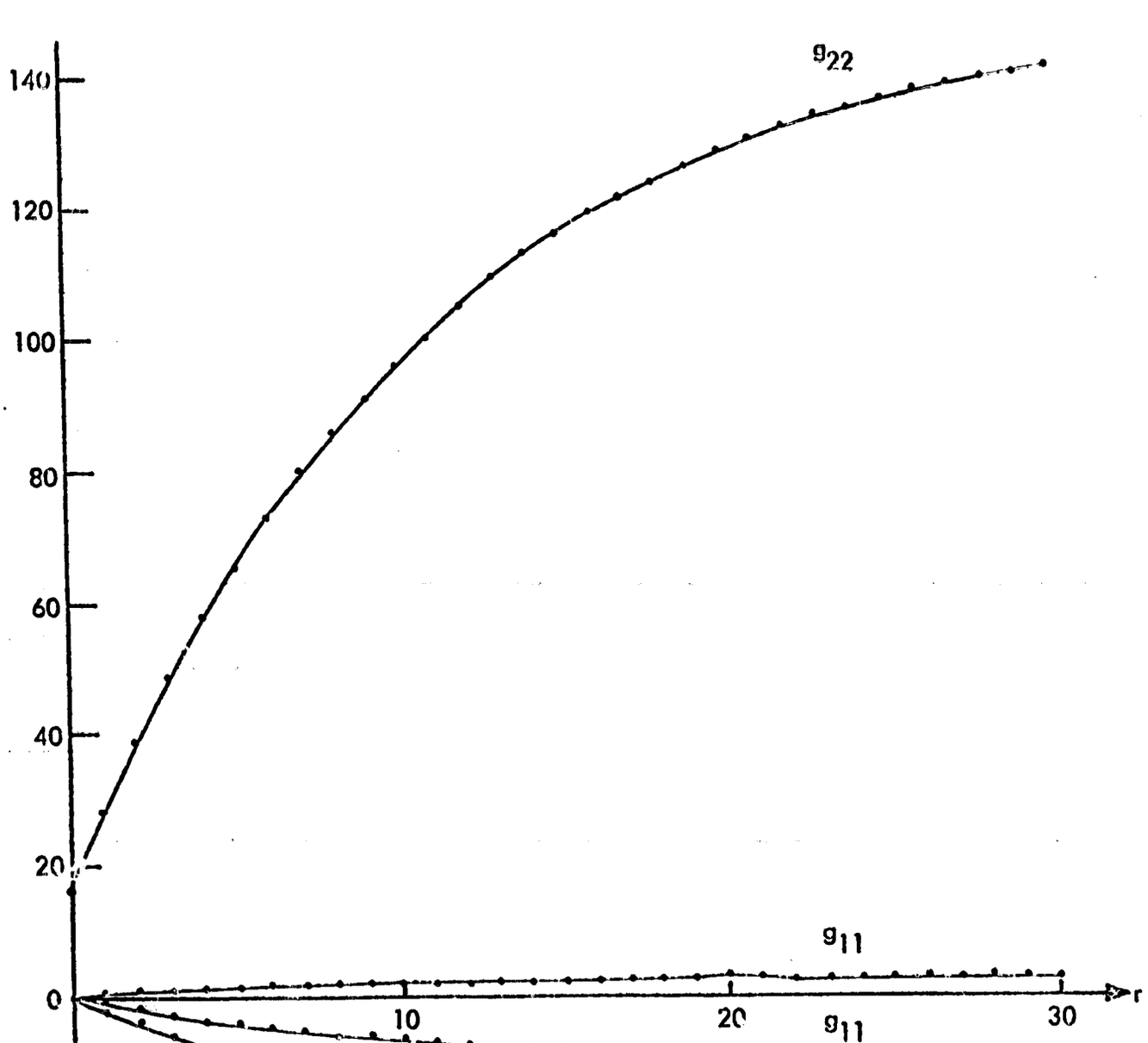


Fig. 5.19 State Step G(r), r = k-θ

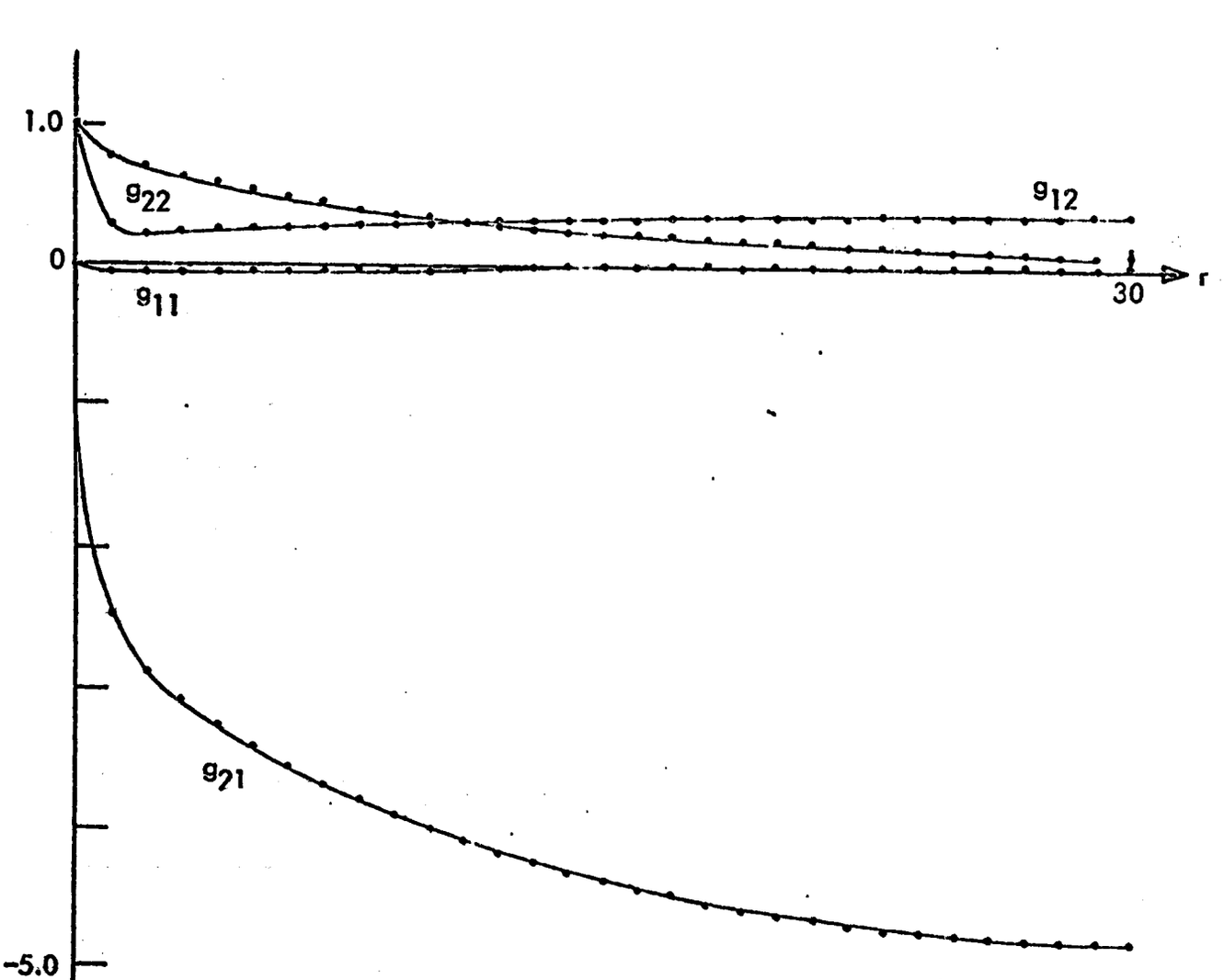


Fig. 5.20 Sensor Step G(r), r = k - θ

$$\underline{\nu}^T = [\nu_1 \quad 0] \; ; \quad \hat{\gamma}_1(k) = G_{11}(k-\theta)\nu_1$$
$$\hat{\gamma}_2(k) = G_{21}(k-\theta)\nu_1$$

and similarly, for angle-of-attack failures

$$\underline{\nu}^T = [0 \quad \nu_2]; \quad \hat{\gamma}_1(k) = G_{12}(k-\theta)\nu_2$$
$$\hat{\gamma}_2(k) = G_{22}(k-\theta)\nu_2.$$

By investigating the relationship between $\underline{\nu}$ and $\hat{\underline{\gamma}}(k)$ on the one hand and the relationship between $\hat{\underline{\gamma}}(k)$ and $\ell(k;\ \theta)$ on the other, one can extract some more information about the degree of detectability of different regions in the failure space. Letting $\underline{\gamma}(j) = \tilde{\underline{\gamma}}(j) + G(j;\ \theta)\underline{\nu}$ in the expression for $\underline{d}(k;\ \theta)$,

$$\underline{d}(k;\ \theta) = \sum_{j=\theta}^{k} G^T(j;\ \theta)V^{-1}(j)[\tilde{\underline{\gamma}}(j) + G(j;\ \theta)\underline{\nu}]$$

$$= \tilde{\underline{d}}(k;\ \theta) + \sum_{j=\theta}^{k} G^T(j;\ \theta)V^{-1}(j)G(j;\ \theta)\cdot\underline{\nu}$$

$$= \tilde{\underline{d}}(k;\ \theta) + C(k;\ \theta)\underline{\nu}$$

where $\tilde{\underline{d}}(k;\ \theta)$ is what would appear if there were no failure at all. Then the log-likelihood ratio can be expressed in a similar fashion:

$$\ell(k;\ \theta) = [\tilde{\underline{d}}(k;\ \theta) + C(k;\ \theta)\underline{\nu}]^T C^{-1}(k;\ \theta)[\tilde{\underline{d}}(k;\ \theta) + C(k;\ \theta)\underline{\nu}]$$

$$= \tilde{\underline{d}}^T(k;\ \theta)C^{-1}(k;\ \theta)\tilde{\underline{d}}(k;\ \theta) + \underline{\nu}^T C(k;\ \theta)\cdot C^{-1}(k;\ \theta)\tilde{\underline{d}}(k;\ \theta)$$

$$\tilde{\underline{d}}^T(k;\ \theta)C^{-1}(k;\ \theta)\cdot C(k;\ \theta)\underline{\nu} + \underline{\nu}^T C(k;\ \theta)C^{-1}(k;\ \theta)\cdot C(k;\theta)\underline{\nu}$$

$$= \tilde{\ell}(k;\ \theta) + 2\underline{\nu}^T\tilde{\underline{d}}(k;\ \theta) + \underline{\nu}^T C(k;\ \theta)\underline{\nu}$$

If we carry this analysis further by studying the incremental variations in $\ell(k;\ \theta)$ considering $\tilde{\ell}(k;\ \theta)$ as a nominal value we could map

out regions in failure space, $\nu_1 \times \nu_2$, which for different times in the windows lead to larger values of $\ell(k; \theta)$ and thus increasing the probability of detection. An analysis of this kind may indicate some a priori limitations in the detection of certain failures. By further investigating these plots of the elements of $G(k; \theta)$ and those of $C^{-1}(k-\theta)$ shown before for the different failure types much qualitative information may be obtained on the performance of the detectors.

Another related area of interest is that of finding approximations to the various curves. Computationally it would be advantageous to be able to replace the $G_{ij}(k; \theta)$, for example, by simple functions such as constants or ramps if detector performance keeps within acceptable bounds. For example, curves of the form

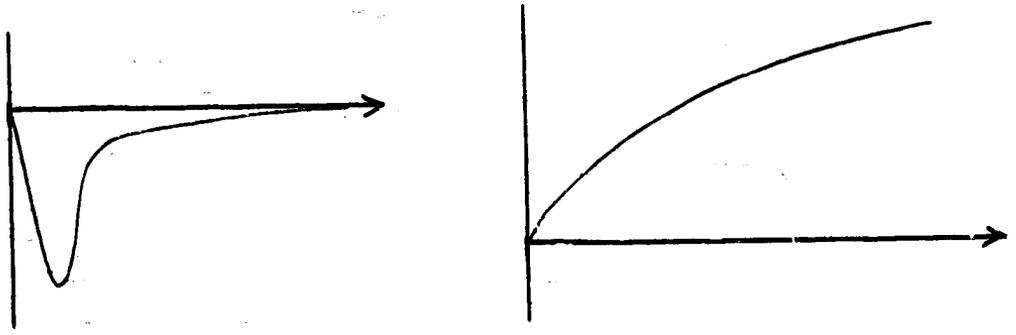


Figure 5.21

might be approximated by simpler ones of the form

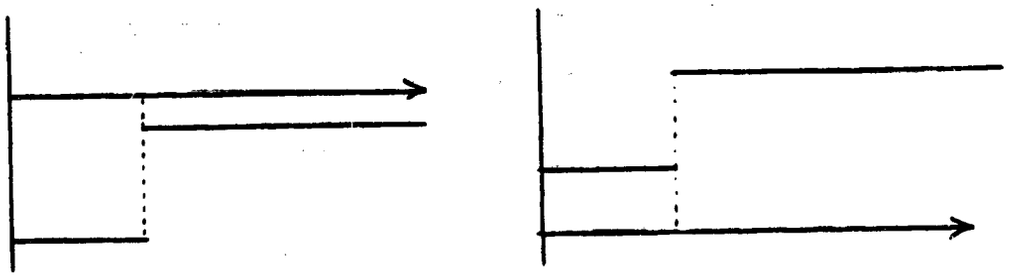


Figure 5.22

or of the form

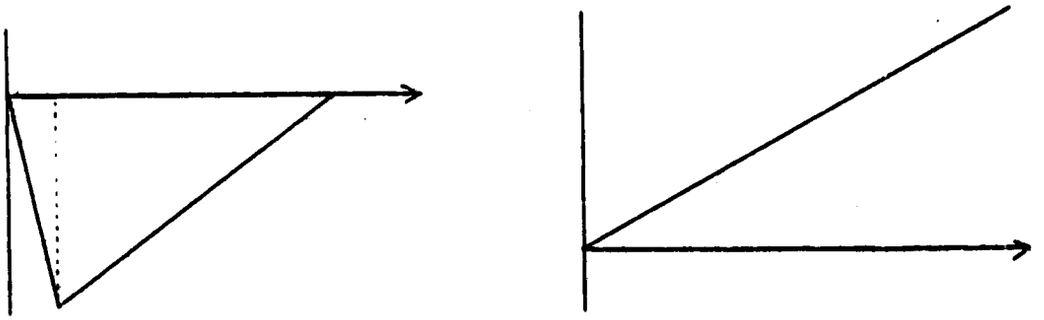


Figure 5.23

The reductions in computation and storage might be significant enough to make the implementation of the full GLR more attractive from a practical point of view. Furthermore, one could try to select the approximating functions in a way such that the sub-optimal design is less sensitive to noise or certain parameter changes, or to accentuate certain features which would render failure classification an easier task once detection has taken place. Alternatively, it may be possible to formulate an optimization problem the solution to which gives the approximating functions, from a specified class of functions, which minimize P_F while maximizing P_D or keeping it constrained to a certain interval ($P_D \geq a$, $0 \leq a < 1$).

Further study of some of the possibilities mentioned seems fruitful in the long run since they offer potentially useful implementation characteristics thus increasing the practicality of this approach. Overall performance might be improved if noise and parameter sensitivity is reduced. When some sensitivity analysis is applied to the detectors it may be interesting to compare the performance to that obtained through the use of the above approximations.

V.7 Directional Effects

Briefly, there are some instances when detector behavior shows some marked differences between data for failures of type $(\nu_1, 0)^T$, in the pitch rate dynamics or observations, and that for failures of type $(0, \nu_2)^T$, in the angle of attack dynamics or its observations. Through the study of such particular features of the performance of the detectors we expect to get some initial information on some sensitivity questions such as flexibility or rigidity of the detector response to changing failures, signal-to-noise ratios for the failures, etc.

The most interesting behavior is captured by Figures 5.24 and 5.25. They show plots of the likelihood ratios in the windows for two different times in the simulations. They are both for sensor steps; Figure 5.24 is for a $20\sigma_q'$ step in the pitch rate sensor and Figure 5.25 is for a $20\sigma_\alpha'$ step in the angle-of-attack sensor. The shape of $\ell(k; \theta)$ as a function of θ for $k = 12$ is seen in Figure 5.24(a) and for $k = 32$ in Figure 5.24(b) for θ inside the window at those times. The true time of failure is $\theta_T = 5$. Notice the monotonically decreasing shape from θ_T on of the $\ell(k; \theta)$ for $(\nu_1, 0)^T$ the failure. Figure 5.25(a) and 5.25(b) show the corresponding data for a failure in the other direction, $(0, \nu_2)^T$.

The difference in the shape of the $\ell(k; \theta)$ is striking. In contrast with the first case shown in Figure 5.24, the second case, shown in 5.25, is like a decaying exponential after θ_T. This phenomenon becomes more noticeable and distinctive as the size of the step failure in the sensor is taken larger. It turns out that the first effect on the $\ell(k; \theta)$ shape is seen in detection of steps in the state in <u>both</u> failure directions. Moreover, the second kind of effect is seen in the shapes of $\ell(k; \theta)$ for

RESIDUALS AT TIME 12
2.2777E-02 -8.0156E-01
GLRS(LG) : 0.8445E+03

**********FAILURE OCCURED AT TIME = 5 = $\hat{\theta}$(k=12)
ESTIMATED FAILURE VECTOR =
1.8366E-01 5.8472E-02

TIME	GLR	GLR BAR GRAPH
1	0.7418E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2	0.7169E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3	0.7494E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4	0.7773E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5	0.8445E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
6	0.7924E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
7	0.7077E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8	0.6697E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
9	0.5766E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
10	0.4262E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
11	0.2689E+03	XXXXXXXXXXXXXXX
12	0.1656E+03	XXXXXXXXX

Fig. 5.24(a)

RESIDUALS AT TIME 32
5.4380E-02 -8.6270E-01
GLRS(LG) : 0.5014E+04

**********FAILURE OCCURED AT TIME = 5 = $\hat{\theta}$(k=32)
ESTIMATED FAILURE VECTOR =
1.7940E-01 2.8816E-02

(b)

TIME	GLR	GLR BAR GRAPH
2	0.4881E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3	0.4914E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4	0.4945E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5	0.5014E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
6	0.4960E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
7	0.4863E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8	0.4825E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
9	0.4716E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
10	0.4557E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
11	0.4422E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
12	0.4301E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
13	0.4155E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
14	0.3962E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
15	0.3790E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
16	0.3654E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
17	0.3440E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
18	0.3289E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
19	0.3083E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
20	0.2918E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
21	0.2695E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
22	0.2486E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
23	0.2263E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
24	0.2047E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
25	0.1815E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
26	0.1614E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
27	0.1422E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
28	0.1249E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
29	0.1013E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
30	0.7712E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
31	0.5538E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
32	0.2307E+03	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 5.24(a,b) GLR's for Sensor Step of 200' in Pitch Rate, (ν_1,0); k=12 in (a) and k=32 in (b).

RESIDUALS AT TIME 12
-5.3951E-02 3.9520E-01
GLRS(LG) : 0.1467E+04 (a)

**********FAILURE OCCURED AT TIME = 5 = $\hat{\theta}(k=12)$
ESTIMATED FAILURE VECTOR =
9.2602E-03 1.2585E+00

TIME	GLR	GLR BAR GRAPH
1	0.1030E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2	0.1060E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3	0.1077E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4	0.1194E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5	0.1467E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
6	0.1083E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
7	0.8095E+03	XXXXXXXXXXXXXXXXXXXXXXXXX
8	0.5563E+03	XXXXXXXXXXXXXXXX
9	0.3814E+03	XXXXXXXXXXX
10	0.2769E+03	XXXXXXXXX
11	0.1882E+03	XXXXXX
12	0.6738E+02	XX

RESIDUALS AT TIME 32
-3.5208E-03 9.1913E-02
GLRS(LG) : 0.1820E+04 (b)

**********FAILURE OCCURED AT TIME = 5 = $\hat{\theta}(k=32)$
ESTIMATED FAILURE VECTOR =
4.8016E-03 1.2288E+00

TIME	GLR	GLR BAR GRAPH
2	0.1136E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3	0.1278E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4	0.1505E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5	0.1820E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
6	0.1437E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
7	0.1162E+04	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8	0.9094E+03	XXXXXXXXXXXXXXXXXXXXXXX
9	0.7299E+03	XXXXXXXXXXXXXXXXXXX
10	0.6232E+03	XXXXXXXXXXXXXXXX
11	0.5420E+03	XXXXXXXXXXXXX
12	0.4168E+03	XXXXXXXXXX
13	0.3553E+03	XXXXXXXXX
14	0.3028E+03	XXXXXXXX
15	0.2574E+03	XXXXXXX
16	0.1991E+03	XXXXX
17	0.1749E+03	XXXX
18	0.1322E+03	XXX
19	0.1135E+03	XXX
20	0.8093E+02	XX
21	0.6752E+02	X
22	0.5496E+02	X
23	0.4515E+02	X
24	0.4194E+02	X
25	0.3361E+02	X
26	0.2755E+02	X
27	0.2582E+02	X
28	0.9529E+01	X

Figure 5.25(a,b) GLR's for Sensor Step for $20\sigma'$ in angle of attack, $(0,\nu_2)$; k=12 in (a) and k=32 in (b).

detections of state jumps, also in both directions. As we try to understand what is going on which leads to these results we expect to clarify further some of the interactions between detection of failures with the GLR method and peculiarities or characteristics of the specific system worked with. Some of the questions in the first paragraph are of interest here.

We can make some remarks here trying to explain the first-order components of this behavior. In Figure 5.19 we saw the elements of the signature matrix for sensor step failures plotted versus $k - \delta$ for the window size considered. Notice that the elements G_{12} and G_{22}, which give the effect of a step failure in α on the residuals, tend to zero while G_{11} and G_{21}, which give the effect of a step failure in q on the residuals, increase and reach non-zero values. Therefore the added components of the residuals due to a step in α begin to diminish immediately after they appear and, not being very persistent, the $\ell(k; \theta)$ for $\theta > \theta_T$ decrease faster than $\ell(k; \theta)$ in the case of a step in q. In the latter case, a step failure in q, the added components of the residuals are persistent, if not increasing, and the $\ell(k; \theta)$ for $\theta > \theta_T$ decrease in value at a lower rate. While there may be other factors affecting the behavior of $\ell(k; \theta)$, they are secondary and modulate these first-order trends.

There are other small details also of similar features which point out differences in detecting failures in particular directions of failure space. For example, in detection of sensor steps in particular, the $\ell(k; \theta)$ are larger in value, on the average, for failures in $(\nu_1, 0)^T$ than for failures in $(0, \nu_2)^T$. It seems to be related to the noise handling capability in each case. In any case, it is necessary to understand the factors at work here, as important questions on the usefulness and

reliability of the GLR detectors are involved and their answers will yield insight.

V.8 Conclusions and Future Investigations

As stated in Section III the purpose of these simulations was to obtain meaningful results with physical interpretation in order to get insight into the dynamics of failure detection via the GLR approach. In the process we have gained some experience with failure detection techniques and some understanding which allows us to evaluate the performance of the detectors being studied. It also helps us to find out what kinds of questions we should be asking as we go along.

A realistic and yet simple model of an F-8 aircraft at a given flight condition was used in simulations in which the detectors corresponding to the various failure modes studied thus far were implemented. The range of failures considered provided us with a large amount of data which displays the basic features of the performance of the GLR detectors.

The qualitative analysis and physical interpretation of the simulation results havealready allowed for an initial evaluation of the observed performance. They have also brought out some of the key factors determining the quality of detection. We expect to further enhance our understanding as we have more time to interpret these results and to integrate them with those to be obtained in the near future.

The overall detection performance seen is excellent for this stage of our research. Detection takes place immediately for most failures considered, certainly for the majority of those of direct interest from an applications point of view. Even those of very small size are still detectable, at least for step failures, although with some degradation in detection performance such as delays to detection time and reduced accuracy of the estimates.

In general, step failures are easily detected and identified and so are jump failures of sufficient magnitude, larger than 1σ for example. In the case of jump failures those in the state are in general much easier to detect correctly than sensor failures. The noise effects are smoothed out by the system dynamics and the jump itself shows up as a persistent disturbance in the measurements. Thus we already anticipate some difficulty in distinguishing between state jump failures and sensor step failures of comparable size in cross-detection studies. Some initial simulation results in which all four detectors are implemented at once for a given failure seem to bear this out. This points to the importance or necessity of using more complex techniques to be able to improve detection and to eliminate redundancy. This was one of the reasons for looking at detection methods such as the GLR approach in the first place. The use of more sophisticated techniques in detection and estimation of failures is undoubtedly necessary when one contemplates compensation in the filter and/or controller for self-reorganizing closed-loop systems. Otherwise one runs the risk of massive and costly system changes without the certainty and accuracy of estimates called for by such action and resulting in degraded, if not unstable, overall system behavior.

The area of decision threshold selection and false alarms must be looked at in more detail. The apparent false alarm rates observed in the simulations are higher than the computed false alarm probability, P_F. We need to study the correlations between the $\ell(k; \theta)$'s with both θ and k varying. For the GLR method presently worked on the log-likelihood ratios are noncentral χ^2 random variables and obtaining their correlations is difficult. This is one of the reasons for studying the simplified GLR (SGLR)

method in which the failure is assumed to be of a fixed form and its estimation is removed from the detection process. For this technique the $\ell(k; \theta)$'s are gaussian random variables which therefore makes such statistical analysis feasible. We hope to derive such quantities as the conditional probability of a false alarm after a false alarm has occurred, the conditional probability of detection after a correct detection and other similar ones of practical and theoretical interest.

The SGLR will be looked at for other reasons as well. The detector equations are being developed for the four failure modes studied so far and the same failures will be simulated as for the full GLR once they are ready. From the results we get we expect to be able to answer some questions on simplifications and sensitivity. The amount of computation and storage necessary for implementing the SGLR method is significantly less than for the full GLR method so that questions arise as to whether the performance will also degrade accordingly. If not, there is the possibility of using the SGLR detectors during normal operation and of switching over to the full GLR once detection takes place. Because of the greater ease and cost-effectiveness of the SGLR one may be able to implement a bank of such detectors for a range of failures of interest in the monitoring phase failure detection. There are many interesting and related questions which remain to be asked as we get more involved in the analysis of the performance of the SGLR method of detecting failures.

One of the next areas to be worked on, which is already under investigation and has been mentioned is that of cross-detection. This refers to the possibility of detecting some or all kinds of failures with a few or one kind of detector only. This is of obvious interest since

one of the drawbacks of the GLR approach as it is now implemented is the fact that a detector is designed for each particular failure mode. Remarks similar to the above ones for the SGLR also apply here except for those about correlations of $\ell(k; \theta)$, a problem which remains in this case since it still is a χ^2 random variable. There are interesting comparisons to be made, once we have some data, on the minimum implementation required to achieve a specified performance in detection as measured in some relevant way.

Another problem that we will be looking at in the near future is what we call detection under mismatched conditions. Briefly, this refers to the question of the actual performance obtained when the real system is somewhat different from the model on which our calculations for the detectors are based. Everything is done as before, but the measurements are made on the simulation of the aircraft at a different flight condition, for example. We expect the results here to tell us something about the sensitivity of the detector performance with respect to changes in parameters of the model which is to represent our knowledge of the real physical system. By doing similar tests on the SGLR detectors we can evaluate the relative robustness of each design to varying conditions. Again these are very important areas that must be investigated since they have many ramifications of interest for applications.

Still further along remains a more complete sensitivity analysis of both the full GLR and the SGLR techniques. We expect to derive some equations for the changes in certain variables of interest such as $\{C(k; \theta)\}_{ij}$, the elements of the information matrix, and P_D and P_F with respect to variations in a parameter vector of importance.

VI. Detailed Description of the Multiple Detector Simulation Program (MDSP)

In order to obtain practical insights into GLR performance, extensive simulation experience is necessary. A single system provides a common basis for comparison and analysis. We are currently focusing our attention on the F-8 aircraft as described in Chapter II. Simulations of a system under different conditions may have a large common data set such as the detector matrices (i.e., $G(k; \theta)$, $C^{-1}(k; \theta)$, etc.). Hence an efficient simulation program must be able to take this into account to eliminate redundant generations of such data sets. In addition, the program should be able to handle different systems sequentially in a single run. Presently, only the four basic detectors (state jump, state step, sensor jump and sensor step) are simulated. However, other detectors such as simplified GLR will be considered soon. Therefore, the program should have enough flexibilities to allow additions. The Multiple Detector Simulation Program (MDSP) is a FORTRAN program which has been developed to achieve these objectives.

The MDSP simulates a system with a single failure and with a set of up to four detectors simultaneously in operation. The MDSP can simulate any number of systems sequentially in each run. For each system, simulations can be performed with different sets of detectors, and for each system and each set of detectors, simulations may be done with a set of different failures, one after another. In order to reduce redundant computations of detector matrices, when detectors are used repeatedly, all detectors are computed before any simulation and stored. They are activated as needed.

Another feature of the MDSP is its ability to simulate mismatched systems. The detectors can be computed according to one system and failure

mode while the simulations are actually of another. This option is aimed at the study of the sensitivity of GLR technique to system parameter variation.

The following section will present a detailed description of the features, structures, function and usage of MDSP. Together with the appendix containing the commented program code and further details in the program algorithm, this section provides a complete description of MDSP. Finally, some possible further additions to MDSP are discussed.

I. The MDSP: Detailed Description

An overview of the program functions is presented in section I.1. I.2 contains the definitions of input data and program usage. Section I.3 describes the structures of MDSP in detail. Precautionary remarks are made in I.4.

Note that this section is intended to provide a clear view of the approach and structure of MDSP and the detailed description of algorithms is not included. The reader is referred to the commented program code in the appendix.

I.1 Program Functions

MDSP simulates linear constant discrete time systems and Kalman filters as described in Chapter 2. Furthermore, the filter is assumed to have reached a steady state; hence the filter gain K(k) becomes a constant matrix K.

The failure modes that MDSP is able to simulate are state jump, state step, sensor jump and sensor step failures numbered type 1, 2, 3

and 4 respectively. The detectors that are based on these failure modes are numbered type 1, 2, 3 and 4 representing state jump, state step, sensor jump and sensor step detectors respectively. More detailed model description of the failures may be found in Chapter 2.

The sequence of operations of MDSP may be traced through by following the function flow diagram (Figure 6.1). It is helpful to bear in mind that MDSP is divided into six blocks (0, 1, 2, 3, 4, 5) each having a different function. After initializations in block 0, the system and filter matrices are read in (block 1). All the detector matrices required in the subsequent simulations are computed according to this system and filter (block 2). If a mismatched system simulation is not chosen, the program proceeds to set up the bank of detectors (the detectors chosen to operate simultaneously during a simulation) in block 3. Otherwise, the mismatched system and filter matrices are read in (block 0), replacing the previous system and filter, before setting up the detector bank. The new system and filter are the ones to be simulated; the old system and filter are the ones the detector system is based upon. Then the simulations of the system, filter and detector bank are performed in block 5 with each of the designated failures inputed in block 4. During the simulation, outputs consisting of detector decision, values of the likelihood ratio, etc., are also provided. After all the simulations of the failures are processed, the program proceeds to determine if another detector bank has been chosen by the user. If so, the new detector bank is set up (block 3) and the cycle repeats. Otherwise MDSP will determine (through data cards) if another system is to be considered. If so, the process after initialization is repeated. Otherwise, MDSP will terminate execution.

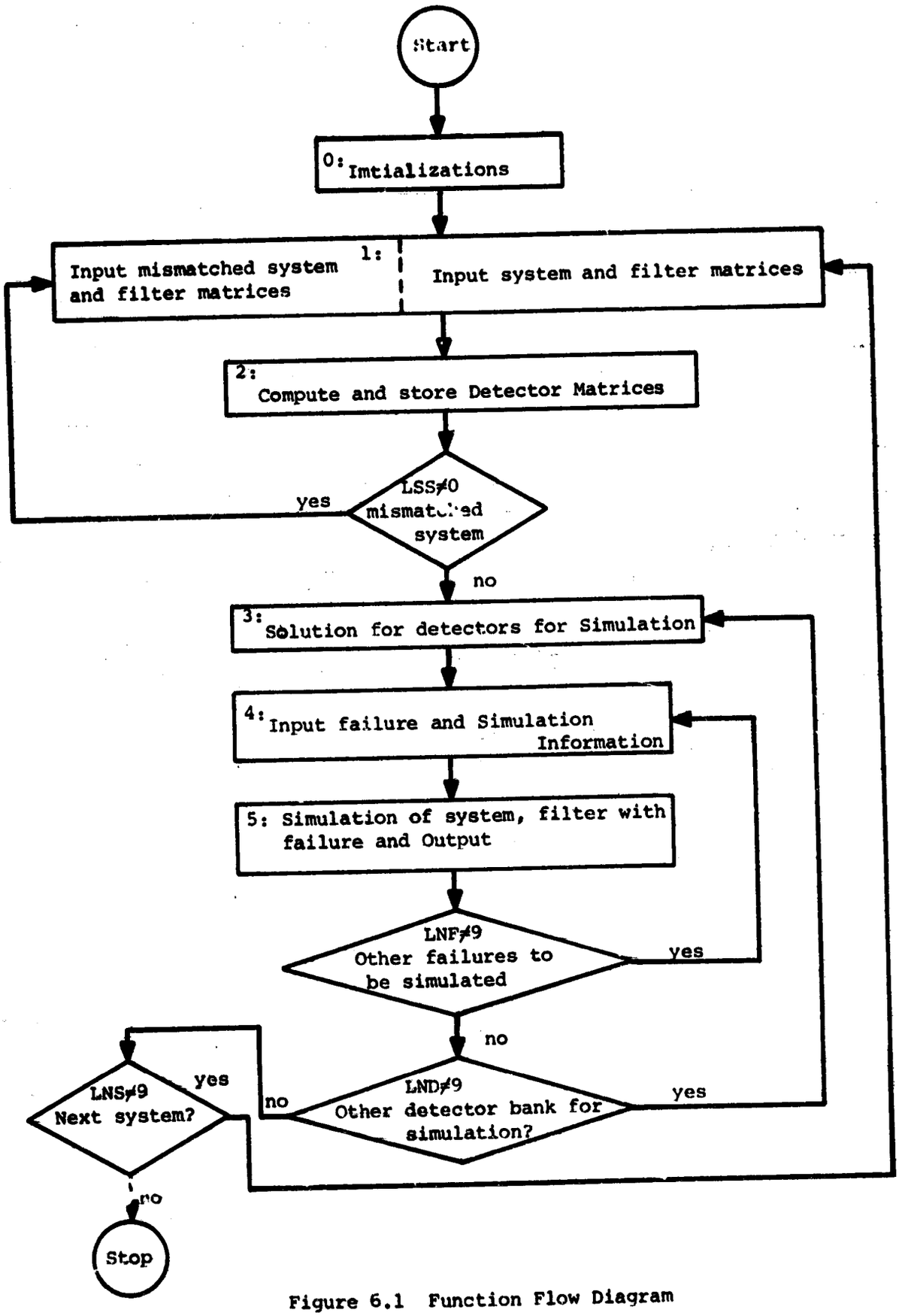


Figure 6.1 Function Flow Diagram

I.2 Program Usage and Input Data

Execution of the program is controlled by variables in the input data set which can be categorized into four subsets:

(I) System and Filter Matrices I/O

LNS - LNS = 9 signals that execution will terminate after the present system has been simulated. Otherwise, the program will proceed to consider a new system after the present one.

IHEAD - An integer array containing the heading or title of the system; its maximum length is 68 characters.

NX - dimension of state vector

NZ - dimension of sensor vector

IOS - I/O control variable of matrices:

IOS = 4 - read and write matrices with title cards

IOS = 5 - read matrices with title cards.

LSS - mismatch control. If LSS = 0, no mismatched system simulation will be performed. A nonzero value indicates a mismatched simulation.

IOM - I/O control variable of matrices of the mismatched system; it takes the same values and meaning as IOS.

These variables are all entered on one card with the following format:

```
      READ (5, 11) LNS, IHEAD, NX, NZ, IOS, LSS, IOM
   11 FORMAT (I1, 1X, 17A4, 2(I2, 1X), 1X, 3I1)
```

In addition, we have the matrix variables:

PHI - Φ

H - H

GNX - G_{NX}

GNZ - G_{NZ}

FK - K

PP - P(k|k-1) in steady state, i.e. $k \to \infty$

P - P(k|k) in steady state

All of these matrices are read in and printed out using the subroutine MATIO. The formats involved are:

Title Card:

READ (5, 1002)

1002 FORMAT (1X, 79H)

WRITE (5, 1002)

Matrix entries:

Each row of a matrix is started on a new card; a row containing more elements than a single card can hold may use as many cards as required as long as the elements are entered consecutively. The format involved is 1000 FORMAT (8E10.0).

(II) Computation and Storage of Detector Matrices

This set of data is entered via a namelist, DETCMP and hence follows the namelist input format.

NUD - number of detectors to be computed and stored.

IOD - [0]*. output control of detector matrices. If IOD = 0, no printing of matrices is done. A nonzero value will cause

the printing of $G(k-\theta)$ for $k-\theta = 0, 1, \ldots M(I)$ (see below for definition of M(I), N(I) and $C^{-1}(k-\theta)$ for $k-\theta = N(I), N(I)+1, \ldots M(I)$. Simple modifications may be made to point out other matrices such as $F(k-\theta)$, $G'(k-\theta)V^{-1}$, etc.

ID - [1] an array containing the detector type number to be computed.

M - [11]. array of the M_i values in the detector window specification.

N - [0]. array of the N_i values in the detector window specification.

{ID(I), M(I), N(I)} specify the type and window of the I^{th} detector in storage.

*[ ] contains the default values if no value is specified on data card. The variables ID, M and N are arrays and each element in these arrays takes a default value as indicated in the brackets.

(III) Detector Selection

This data set is also entered via a namelist, DETSZL, which specifies the detector bank for simulation.

LND - [0]. LND = 9 indicates the end of simulation of the present system. Otherwise, a new detector bank is chosen for simulation and it is specified in this namelist.

IDS - array containing the detector types in the bank.

MS - [11]. array of the M_i values of the detectors in the bank.

NS - [0]. array of the N_i values of the detectors in the bank.

EP - [6]. array of CLR threshold values for the detector in the bank.

{IDS(I), MS(I), NS(I), EP(I)} specifies the I^{th} detector in the bank. The detector bank should only contain detectors that are in storage and possibly with windows that are smaller than the corresponding ones in storage but these windows must be contained in the ones in storage.

(IV) Failure and Simulation Data

This set is entered via the namelist FASIM.

LNF - [0]. LNF = 9 indicates the end of simulation with the present detector bank. Otherwise, simulation is to be performed with the true failure specified in this namelist.

IOP - [2]. Output option for detector decision; it can take on possible values (1, 2, 3). The options are explained in the next section under block 5.

IFL - [1]. the type of failure to be simulated.

KTF - [3]. the true failure time (≥ 1)

NKM - [15]. time after which simulation is to stop.

RNU - [0]. failure vector ν (of type ILF).

(V) Captions

This data set is constant and is read in only once and before all other data. It consists of two integer arrays, ITYPE and IDET which contain characters that are to be printed at appropriate places of the output as captions.

ITYPE - a 5x4 array with contents:

locations	contents
ITYPE(I,1) for I = 1,2,...5	JUMP IN STATE AT
ITYPE(I,2) for I = 1,2,...5	STEP IN STATE AT
ITYPE(I,3) for I = 1,2,...5	JUMP IN SENSOR AT
ITYPE(I,4) for I = 1,2,...5	STEP IN SENSOR AT

IDET - a 5x4 array with contents:

locations	contents
IDET(I,1) for I = 1,2,...5	STATE JUMP DETECTOR
IDET(I,2) for I = 1,2,...5	STATE STEP DETECTOR
IDET(I,3) for I = 1,2,...5	SENSOR JUMP DETECTOR
IDET(I,4) for I = 1,2,...5	SENSOR STEP DETECTOR

Under this setup, the sub-array of ITYPE with a failure type number as the constant second array index contains the description in words of the failure type. As a result, the failure description is indexed by the failure type number. IDET and detector type number have the same relationship as ITYPE with failure type number.

For the present version of MDSP, detector type as well as failure type can take on four values: 1, 2, 3, 4 representing the four basic types: state jump, state step, sensor jump and sensor step respectively. Note that the main program control variables are the flags LNS, LSS, LND and LNF.

To illustrate the ordering of these input data, the following example is included.

I.3 Program Structure

MDSP is divided into six blocks that interact with one another much like subroutines. The subroutine approach is not employed due to the need to pass a large number of arguments and the number involved. The blocks are separated clearly both functionally and physically so as to facilitate the understanding of the program. The one disadvantage is that attention must be paid to variable names so that they are not used for different quantities in future alterations of the program.

The six blocks divide the code according to the following conventions:

	statement numbers		block function
Block #	from	to	
0	0	999	initializations
1	1000	1999	system and filter matrices I/O
2	2000	2999	compute and store detector matrices
3	3000	3999	detector selection
4	4000	4999	input failure and simulation data
5	5000	5999	simulation and output results

There are no intermingling and overlapping statements from different blocks and there are comment statements separating the blocks and stating their functions. All format statement numbers contain at most two digits with the most significant digit indicating the block number in which the format is first used and defined.

The following is a description of the block functions.

(I) Block 0 - Initializations

This block initializes some interval variables of the program. (For details consult comments in program code). The caption arrays ITYPE and IDET are initialized via read statements.

(II) Block 1 - System and Filter Matrices I/O

This block provides the code for reading in and printing out system and filter matrices as well as the I/O for the mismatched system and filter. In fact, the same code and storage are used for the original and mismatched matrices. If the mismatch option is used, the original system matrices are erased after the detectors are calculated.

(III) Block 2 - Computation and Storage of Detector Matrices

As the execution of this block begins, the data namelist DETCMP is read in. By detector (matrices), it is meant the matrices $F(k - \theta)$, $G(k - \theta)$, $G'(k - \theta)V^{-1}$, $C(k - \theta)$ and $C^{-1}(k - \theta)$. Even though not all of these matrices are used in the other blocks in the present version of MDSP, all of them are stored, anticipating future additions that would utilize them.

The G matrices of a detector are ordered in increasing value of $k - \theta$ and stored in a section of a big storage area for the G matrices of all the detectors chosen to be stored. Pointers are created to indicate the beginning of the sections for different detectors. Other detector matrices are likewise stored. The term "pointer" has the meaning of an "offset" throughout the content of this documentation. Rather than pointing directly to any section, the pointer provides the storage space to be skipped from the very beginning of the big storage to get to the

section. We have the following relation of storage and pointers.

matrix type	name of the single matrix in program	name of the big storage	pointers associated with the storage
$F(k-\theta)$	F	FS	LF
$G(k-\theta)$	G	GS	LG
$G'(k-\theta)V^{-1}$	GTVIN	GVS	LV
$C(k-\theta)$	C	CS	LC
$C^{-1}(k-\theta)$	CINV	CIS	LC

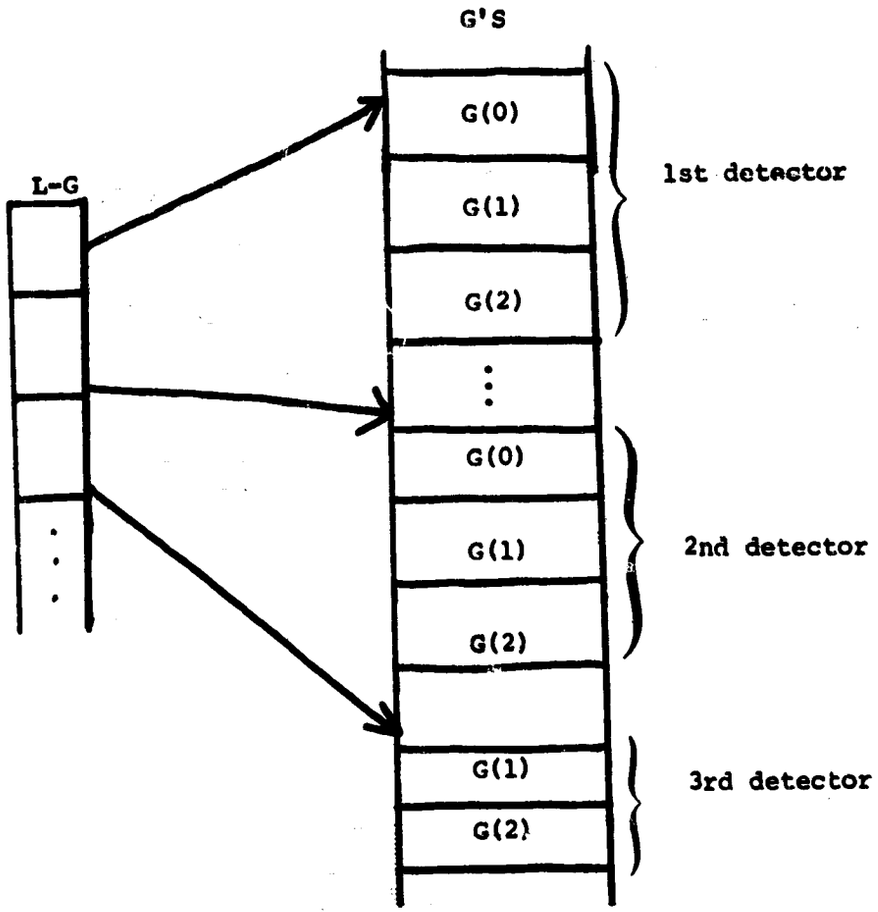
Note that the pointers are in fact arrays.

Recall that the dimension of the G matrix depends on the type of failure assumed, (NZ) x (NX) for state failures and (NZ) x (NZ) for sensor failures. Hence, the second dimension varies from detector to detector and is stored in an array LNZ. Then for the I^{th} detector, we have the following specifications and pointers:

{ID(I), M(I), N(I), LNZ(I), LG(I), LF(I), LC(I), LV(I)}.

The sizes of the sections in storages may be determined as follows:

Storage	size of I^{th} section
FS	(NX) * (M(I) + 1)
GS	(NZ) * (M(I) + 1)
CS, CIS	LNZ(I) * (M(I) - N(I) + 1)
GVS	LNZ(I) * (M(I) + 1)


G'S
L-G
G(0)
G(1)
G(2)
1st detector
G(0)
G(1)
G(2)
2nd detector
G(1)
G(2)
3rd detector

According to the previously defined nature of the pointers, we have

$$LG(1) = LF(1) = LC(1) = LV(1) = 0$$

The value of the (I+1)st pointer is equal to the value of the Ith pointer plus the size of the Ith section.

For nonzero values of IOD, GS and CIS are printed. However, other storages may also be printed with some very simple addition to MDSP.

(IV) Detector Selection

The namelist DETSEL is read in at the beginning of the block.

Since all the detectors have been computed and stored, any selection of these detectors only involves the correct identification of pointers. For each value in IDS (indicating the detector type) the array ID is searched for the same value. Then corresponding values in the pointer arrays set up in block 2 are assigned to a new set of pointer arrays. Consider the Ith detector in the detector bank. If J is the index of ID such that

$$IDS(I) = ID(J)$$

then we set

$$JG(I) = LG(J)$$

$$JF(I) = LF(J)$$

$$JC(I) = LC(J) + (NS(I) - N(I)) * LNZ(J)$$

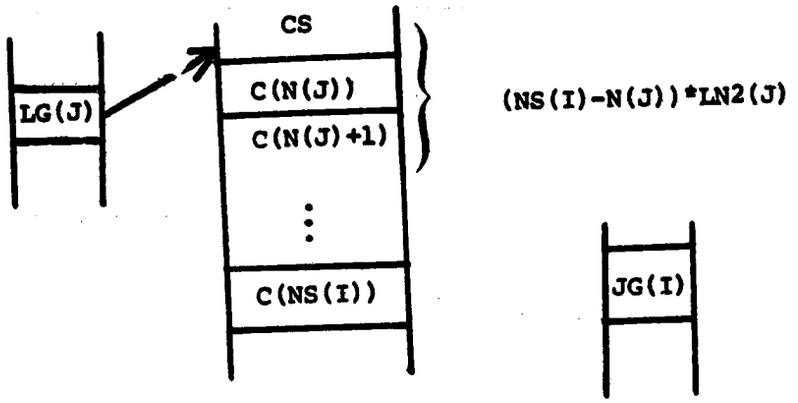
$$JV(I) = LV(J)$$

$$JNZ(I) = LNZ(J)$$

If IDS(I) does not have a matching value in ID or [MS(I), NS(I)] is not compatible with [M(J), N(J)], i.e. MS(I) > M(J) and/or NS(I) < N(J)

then IDS(I) is dropped and the following values in IDS, MS, and NS are moved up in the arrays while a message is printed indicating such incompatibility. Then NUDS is also decremented by 1.

Since we allow NS(I) $\geq$ N(J), the first C matrix in CS we want considered is NS(I) - N(J) matrices after the beginning of the section. Hence, JC(I) = LC(I) + (NS(I) - N(J)) * LNZ(J).



In addition, two more pointer arrays are created: JR and JD. The former points to GLRS (storage for $\ell(k;\theta)$ of all the detectors) and the latter points to DTS (storage for $d'(k;\theta)$ of the detectors). The nature of JR and JD are similar to the other pointers. The sizes of the Ith sections in GLRS and DTS are MS(I) - NS(I) + 1 and MS(I) + 1 respectively.

The detector bank specifications (IDS, MS, NS) are printed for eacy reference.

(V) Block 4 - Input Failure and Simulation date

Here the namelist FASIM is read in and information contained in FASIM is printed out for easy reference.

(VI) Block 5 - Simulation and Output

This block consists of four sub-blocks: (a) system and filter simulation, (b) generation of $\ell(k; \theta)$ for the detectors in the bank, (c) detector decisions, and (d) output.

(a) System and filter simulation

The system is simulated with the failure specified in block 4. The filter generates the residual necessary for the detectors. The algorithms involved are simple and the code is self-explanatory.

(b) Generation of $\ell(k; \theta)$

The computation of $d(k; \theta)$ and $\ell(k; \theta)$ is straightforward. However, the manipulation of quantities in the storage DTS is involved and is explained in detail in the appendix. This sub-block generates the $d(k; \theta)$'s and $\ell(k; \theta)$'s for the detectors in the bank via subroutine calls to GENDLR.

(c) Detector Decision

The GLRS array is examined to determine if any $\ell(k; \theta)$ is greater than the thresholds (EP). The number of $\ell(k; \theta)$ crossing the threshold is recorded for each detector and the five times (values of θ) that have the largest $\ell(k; \theta)$ exceeding the threshold for each detector are ordered in decreasing values of $\ell(k; \theta)$.

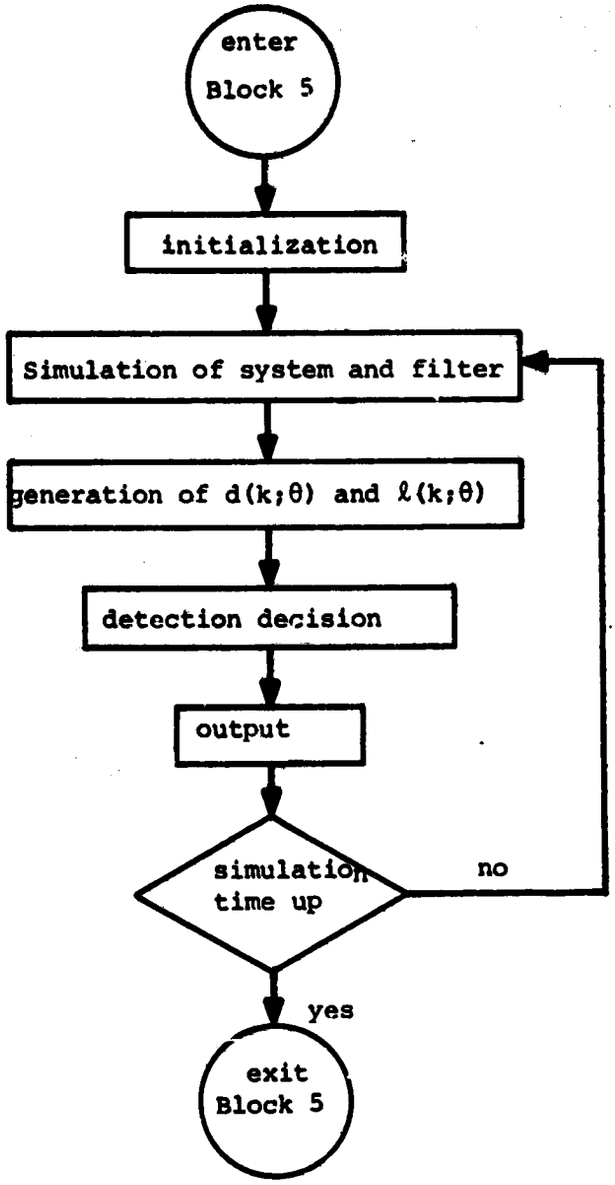
A maximum likelihood estimate (MLE) of the failure vector is made for the θ the has the largest $\ell(k;\theta)$ by each of the detectors that detects a failure (i.e. having some $\ell(k;\theta)$ exceeding the threshold).

(d) Output

Presently there are three output options all of which deal with the plotting of GLR bar graphs. At each time step, the time and the Kalman filter residual γ are printed. For all three options, if no $\ell(k;\theta)$ crosses the threshold for a detector, nothing is printed for that detector. Otherwise the following is printed for the detector:

1. detector number - is the number J such that IDS(J) specifies the present detector and hence is not the type number. This is done because the detector bank may contain two detectors of the same type but having different windows. Then the type number is not a good identification. (The correspondence between a detector, its description (type window and threshold) and its detector number in the bank were pointed as block 3 was executed).
2. the largest $\ell(k;\theta)$ value.
3. the five values of θ having the largest $\ell(k;\theta)$ exceeding the threshold ordered in decreasing values of $\ell(k;\theta)$
4. MLE of the failure vector (RNUE is the estimate in the program).

Block 5: Simulation and Output



For the bar graph output:

Option 1 (IOP=1) - no ploting of bar graph.

Option 2 (IOP=2) - plot $\ell(k;\theta)$'s for all detectors in simulation if any detector detects a failure.

Option 3 (IOP=3) - plot $\ell(k;\theta)$'s for all detectors in bank at all times.

For a detector that does not signal a failure (no $\ell(k;\theta)$ exceeds threshold), the bar graph of $\ell(k;\theta)$ is scaled as the threshold equivalent to full scale. Otherwise, the largest $\ell(k;\theta)$ is the full scale value.

I.4 Words of Caution

MDSP manipulates a large number of arrays. Hence, much attention should be paid to the dimensioning of arrays to avoid painful error such as writing over onto other arrays. Rules and advices may be found in the comments in the code.

Many of the matrix manipulations in the program employ subroutines developed at the Electronic Systems Laboratory (ESL), M.I.T. (e.g. MATIO, MMUL, MAT4, etc). The ESL has also developed discretization and Kalman filter gain computation packages which were used in obtaining the system and filter matrices of the second order F-8 in our simulation studies. MDSP is compatible with the above packages.

The subroutine, GAUSS, used by MDSP to generate random numbers is a subroutine from the IBM scientific package.

II. Flexibility for further Modules

The block structure of the program has no complicated inter-block interactions. This facilitates the additions of further modules as new blocks. The lack of intertaining logic in MDSP (except in Block 5 where the $d'(k;\theta)$ vector are generated and where detector decisions are made; but both of these processes are self-contained units) makes the addition of sub-blocks simple.

Presently the addition of simplified GLR detectors to MDSP is being considered. Computation of the two probabilities, P_D and P_F of the detectors and simulation of multiple failures are examples of possible additions and expansions of MDSP. Furthermore, MDSP may be easily modified to accept residuals of the Kalman filter from an external source, making it possible to be used in conjunction with, for instance, a non linear simulation.

VII. Future Work

Based on the results outlined in this report, we plan to consider the following issues.

Task #1: Cross-Detection and Wrong-Time Probability Calculations. We plan to apply the techniques outlined in Section II to the simplified F-8 model.

Task #2: Measure of Failure Mode Indistinguishability. Motivated by the cross detection problem, we are planning to develop a measure of mode indistinguishability. Our initial attempts will involve the use of inner products of failure signatures and the definition of "orthogonal failure modes." We hope to develop a Gram-Schmidt Orthonormalization Procedure for a set of failure modes. The idea here is to determine a transformed set of signatures so that only one likelihood ratio will become large when any particular failure occurs. This will greatly simplify the resulting detector decision logic. In addition, this study can lead to the determination of a small set of "universal signatures," which can be used to detect a wide variety of failure, the idea being to use these signatures to detect failures with subsequent isolation provided by correlating residuals with a larger set of signatures.

Task #3: Sequential Probabilities for SGLR. In Sections II and IV, we noted that if one utilized the "window" approach to GLR, one would need to calculate joint probabilities that likelihood ratios $\ell(k;\theta)$ exceed some

threshold. In addition, at the end of Section IV, we proposed a possible detection scheme that requires K likelihood ratios to exceed a threshold. By doing this, we may reduce false alarm problems substantially. Recall that in Section V we observed that false alarms occurred in bursts -- i.e. one bad data point would successively trigger off a number of alarms. The approach outlined in Section IV would help minimize this problem. We also note that, as described in Section V, the shape of the set of $\ell(k;\theta)$ after a failure is quite distinctive and is much less so when false alarms occur. Thus, it is clear that a study of the correlation behavior of the LR's would be extremely useful in allowing one to utilize the GLR data in an optimum manner. However, as mentioned in Section II, the LR's in the full GLR case are noncentral χ^2 variables, and the study of correlated variables of this type is quite difficult. Thus, we propose to study these questions for SGLR, where all of the variables are Gaussian. We feel that SGLR is "close enough" to GLR so that our analysis will be valid (in general terms) for full GLR as well. Thus, we plan to examine the sequential correlation of SGLR's and to use this information in the development of efficient detection rules. In addition, one of our first uses of this information will be in the calculation of delay time in detection -- i.e. the calculation of

$$\text{Prob}\left\{ \ell(k;\theta)>\varepsilon \,\middle|\, \begin{array}{l} \ell(\theta;\theta),\ldots,\ell(k-1;\theta)<\varepsilon; \text{ and a failure} \\ \nu \text{ occurred at time } \theta \end{array} \right\}$$

We feel that this task will provide some of the most useful tools for future development of the GLR technique.

Task #4: Cross Detection Simulation Studies. We plan to run a series of simulation runs using the simplified F-8 model in order to study the qualitative properties of GLR cross detection. We hope that these results, together with the indistinguishability results from Task #2 and the analytical tool of Task #3 will allow us to develop a method for minimizing the cross detection effect.

Task #5: Sensitivity Studies and Simulations. One of the key unanswered questions is the robustness of GLR to model errors. We plan to run a series of simulations in which the GLR is designed based on the linearized F-8 dynamics at one flight condition, while the plane is actually at a second condition.

Task #6: Simulations of SGLR. It has been conjectured that SGLR, while not as accurate a detector as GLR, might be far less sensitive to parameter errors. We plan to implement a set of SGLR simulation routines and to run a series of simulations in order to study the utility of this method.

Task #7: Development of Several Pedagogical Examples. The GLR approach as we have been developing it has a number of aspects that are somewhat subtle. We feel that in order to clarify these issues, it would be nice to have several very simple test problems that make the various points relatively easy to observe. We have already done this once (in Section V) and will attempt to find several other test problems that illustrate key points.

There are clearly a number of other issues that must be considered in this study; however, the above represent those tasks we plan to complete during the present grant period. A full set of additional tasks will be spelled out in the proposal for continuation of the grant.

REFERENCES

1. W.H. Lee, K.-P. Dunn, M. Athans, "Reduced State Designs for the F-8 Aircraft Dynamics," Interim Report 9 for NASA Langley Grant NSG-1018, M.I.T. Electronic Systems Lab., Cambridge, Mass., Oct. 15, 1975.

2. E. Chow, K.-P. Dunn, and A.S. Willsky, "Research Status Report to NASA Langley Research Center: A Dual-Mode Generalized Likelihood Ratio Approach to Self-Reorganizing Digital Flight Control System Design," M.I.T. Electronic Systems Laboratory, Cambridge, Mass., April 1975.

3. A.S. Willsky, "A Survey of Deisgn Methods for Failure Detection in Dynamic Systems," M.I.T. -NASA/Ames Workshop on System Reliability Issues for Future Aircraft," M.I.T., Cambridge, Mass., August 18-20, 1975; also submitted to Automatica.

Appendix

I. Algorithms in MDSP

All the algorithms used in the MDSP are straighforward except the generation of $d(k;\theta)$ and $\ell(k;\theta)$ in block 5. The computation of $d(k;\theta)$ is performed in the subroutines GENDLR called in block 5 according to the formulas:

$$(1)\quad d(k+1;\theta) = d(k;\theta) + G'(k+1-\theta)V^{-1}\gamma(k+1)$$

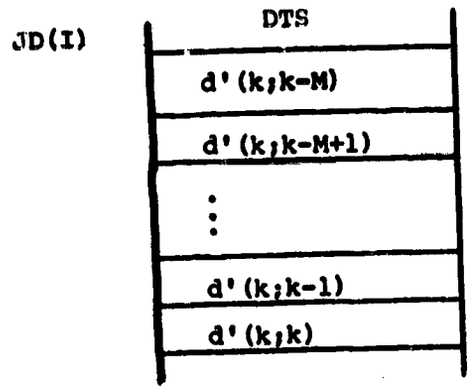
$$(2)\quad \ell(k+1;\theta) = d'(k+1;\theta)C^{-1}(k+1;\theta)d(k+1;\theta)$$

$d(k;\theta)$ may be expressed alternately as:

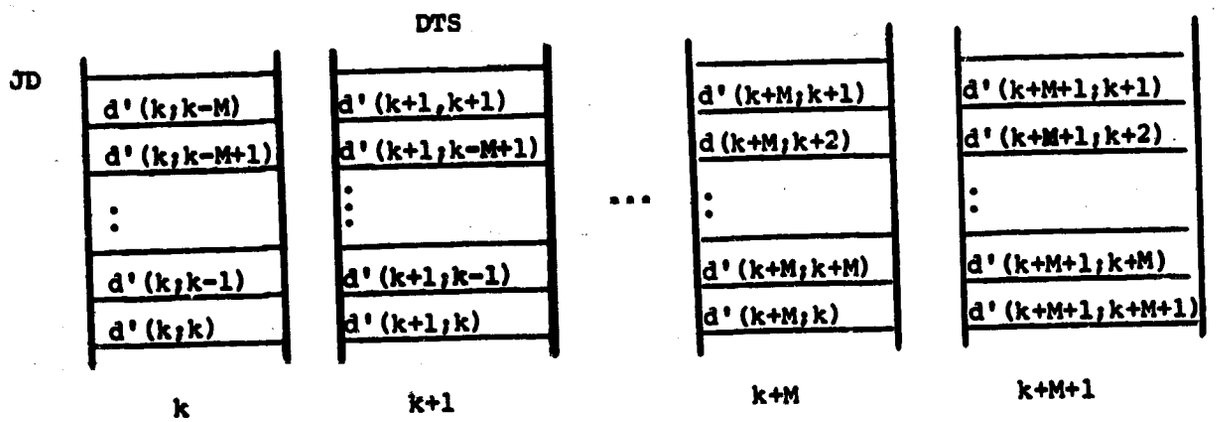
$$(3)\quad d(k;\theta) = \sum_{j=\theta}^{k} G'(j-\theta)V^{-1}\gamma(j)$$

For a feasible realization of the GLR detection scheme, a finite size window characterized by $k-M\leq\theta\leq k-N$ is used. The computation of $d(k;\theta)$ requires $k-\theta+1$ data points and in particular, $d(k;k-M)$ requires $M+1$ data points from $k-M$ to k. Consequently, the actual window $k-M\leq\theta\leq-N$ implies an affective window $k-M\leq\theta\leq k$ within which all the d's have to be stored to utilize formula (1). (Alternatively, $M+1$ γ's would have to be stored to use formula (3)).

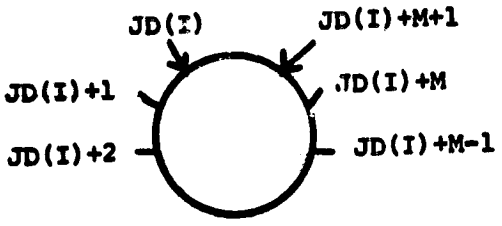
There are several ways to store the d's in the effective window to conserve storage. The scheme employed in MDSP is as follows. Consider the section of DTS (storage of d's for the whole detector bank) for the I^{th} detector in the bank:



At time k+1, d(k;k-M+1) through d(k;k) may be incremented to form the M d's for the effective window at k+1 via the recursive formula (1). However, d(k;k-M) is not used since its incrementation gives d(k+1;k-M) indicating an observation point outside the effective window into the past and hence a point outside the actual window. But this storage space is now used for d(k+1;k+1), a new point in the effective window acquired through the sliding of the window. Continuing this "replacement" process, the same section of DTS at various times contains:

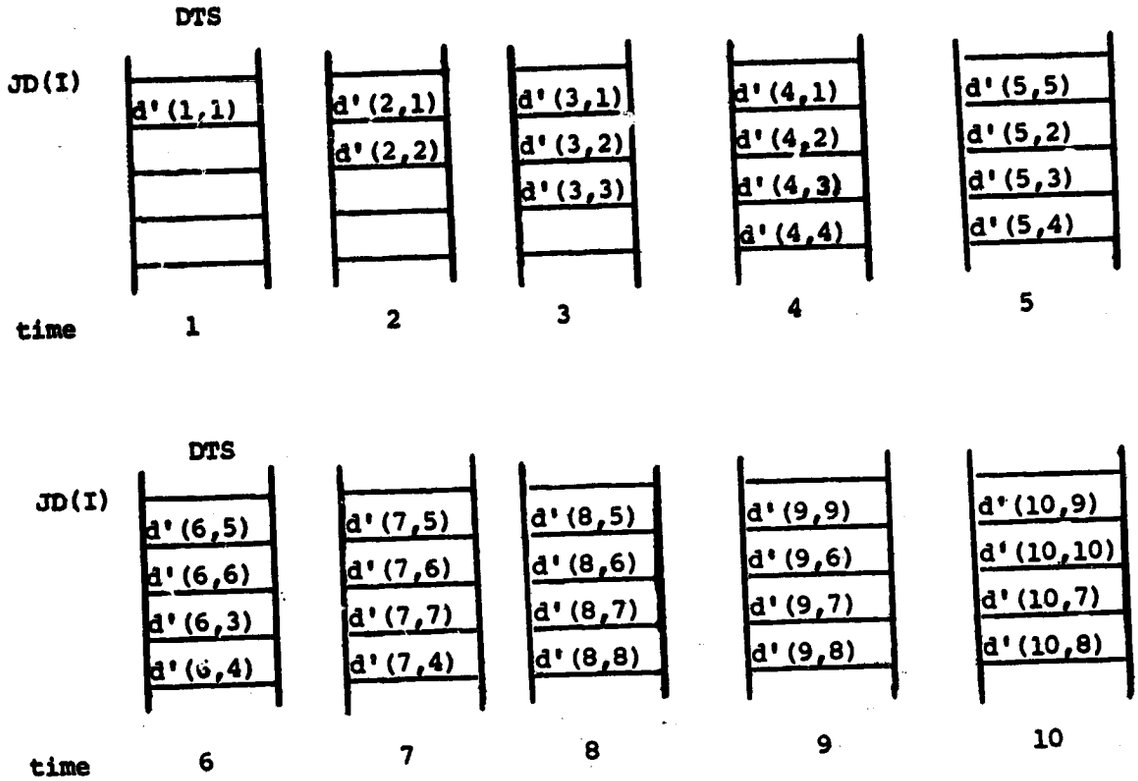


Effectively we have treated this section of storage as "closed-end" or "circular" storage.



To be able to tell where the ends of the effective window are, we defined another offset array JS. [JD(I) + JS(I)] points to the farthest point into the past in the effective window. Hence JS(I) is an offset from the pointer JD(I). Other points in the window may be easily located treating the storage section as a "circular" storage.

As each detector is activated, there is a transient during which the effective window is not filled i.e. there is an m such that $0 \leq m \leq M$ and $\theta = k-M < k_0$, the starting time. Hence, during the transient, the effective windows is smaller than it is later. As time progresses, the effective window grows until it reaches its full size (M+1). Therefore, we define an array JM to record the sizes of the effective windows of the detectors in the bank from the starting time on. The effect of JS and JM on the storage section may be best visualized via a diagram. For simplicity, we let M=3.

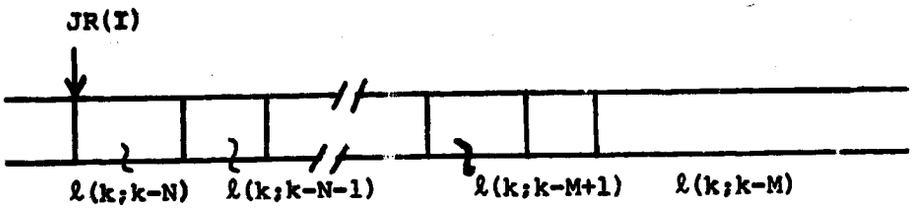


← : JD(I) + JM(I)

—o : JD(I) + JS(I)

Note that JM(I) is set to M for $k \geq M+1$. After the transient (M+1 time steps), JS(I) is reset to zero every M+1 time steps simulating the "circular" effect of the storage.

For each new set of d's, a set of ℓ's is computed for the actual window (i.e. $k-M \leq \theta \leq k-N$) and stored in the GLRS array in the following manner:



Except the updating of JS and JM as time progresses, the process described in this section is carried out by the subroutine GENDLR.