International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

Detection of Changes in Dynamical Systems with Application to Mathematical Immunology

Kolodziej, W.J.

IIASA Working Paper

WP-89-067

September 1989

WORKING PAPER

DETECTION OF CHANGES IN DYNAMICAL SYSTEMS WITH APPLICATION TO MATHEMATICAL IMMUNOLOGY

W.J. Kolodziej

September 1989
WP-89-067

IIASA
International Institute
for Applied Systems Analysis

DETECTION OF CHANGES IN DYNAMICAL SYSTEMS WITH APPLICATION TO MATHEMATICAL IMMUNOLOGY

W.J. Kolodziej

September 1989
WP-89-067

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Foreword

A method is developed for the detection of parametric change for conditionally-linear stochastic systems. Such systems are quite prevalent in biology, economics and engineering, and may be synthesized as certain bilinear stochastic systems with output feedback (possibly nonlinear) through the controls. In other words, these systems are linear in the "unmeasured states" and nonlinear in the "measured states". The previously-derived conditionally linear filter forms a convenient part of the algorithm which estimates the time of change and the parameter values.

The particular motivation here is for the application to immunology and clinical practice. In this regard, a simple example is presented.

A.B. Kurzhanski
Chairman
System and Decision Sciences Program

DETECTION OF CHANGES IN DYNAMICAL SYSTEMS WITH APPLICATION TO MATHEMATICAL IMMUNOLOGY

W.J. Kolodziej

Introduction

The problem of detecting changes in dynamical properties of signals and systems is studied here. The need for this study is motivated by the following:

- modeling of systems and signals where the structural decomposition takes place [1],
- monitoring a dynamical system for abrupt changes of its parameters (e.g. failure of its functions or components) [2],
- complementing an adaptive identification procedure in case of non-smooth parameter changes [3].

The particular motivation for this study stems from specific problems encountered in modeling immunological systems. Such systems are by nature very complex and the corresponding models are at best a crude approximation to the actual biological mechanisms. Very often a poor understanding of these mechanisms does not allow accurate modeling. Even if such mechanisms are understood the lack of consistent data does not allow reliable model identification and verification. Therefore, immunological models must be at all times monitored for validity. If such models are the basis for decisions affecting a particular biological system then the discrepancies between the model (which is usually describing average properties) and the individual system responses must be detected and identified. In such detection one should be able to distinguish between statistical outliers and the errors resulting from model oversimplification. Assuming that the models will always be to a certain degree invalid, one of the tasks for the detection algorithm is to monitor the level of "invalidity" and signal only appreciable differences. The robustness of detection methods must be also a primary concern because the statistical properties of biological experimental data are poorly known.

A growing attention to detection problems has been seen for the last two decades. The comparison of review papers by [2], and [4] shows a significant shift towards sophisticated statistical tools while still preserving elements of system theory. In particular the so-called asymptotic local approach, with the foundations in Le Cam's work [5] on contiguity of probability measures, provides here the theoretical framework for statistical sequential testing. The sequential approach is of main interest if the change detection

algorithm is to assist the on-line decision process (e.g. control law or model acceptance). Among the problems which did not receive adequate attention are:

- robustness of the detection methods with respect to the violated assumptions about the mathematical/statistical properties of underlying models,
- implementation aspects of the detection algorithms and their performance with respect to standard statistical methods,
- compromise between generality of the proposed approach (and accordingly the level of mathematical abstraction) and the actual level of mathematical knowledge of the engineering community.

As an example of the last concern the theory of mixingales ("asymptotic" martingales) is used very often in the study of probabilistic measures convergence. Such mathematical concepts are not known by the majority of applied mathematicians and engineers. In order for the methods based on such theory to become popular they have to significantly outperform standard statistical techniques which are relatively intuitive and readily understandable. This is particularly important as there is in general limited feedback between the engineers or scientists who are responsible for algorithm implementation and the applied mathematicians who provide mathematical framework for such algorithms. The ability of "tuning" and improving the proposed techniques must be in the hands of people responsible for implementation and "maintenance" of detection algorithms.

In this paper an attempt to combine already classical results with some novel ideas is made with the issues discussed above taken into account. Use of continuous time is exercised, however some of the results presented here for continuous stochastic processes are merely conjectures. The insights gained by using continuum allow the development of the discrete time methods which have a solid mathematical base. We do not consider discrete time as the only practical approach in actual algorithm implementation. On the contrary an efficient use of computers requires an insight into algorithm structure, which the discrete time domain seems to obscure (lengthy and unwieldy formulas etc.). Also not without significance is the fact that system modeling starts very often from the physical laws which use continuous time. This is particularly well seen in mathematical immunology where continuous time models dominate [6], [7]. The stochastic aspects of such models come not only from the noisy measurements but also from the randomness associated with immunological processes (e.g. antigen sources and growth, cell stimulation and differentiation) [1].

A simple example of the detection technique which deals with continuous models and discrete measurements is also presented. This example illustrates some of the concepts presented here and provides the basis for further experimentations.

1. Continuous Time Detection

We begin with the discussion of change detection in the continuous time processes and systems. The theoretical difficulties encountered here do not allow a complete design of the detection algorithms but only provide insight into their mathematical structure. It seems natural to analyze the behavior of continuous time random dynamical systems since the immunological processes are well defined from conservation equations and chemical mass-action principles governed by the sets of differential equations [6], [7].

Assume that we observe (i.e. measure realizations of) a process y_t which satisfies an Ito type equation. This models in a formal manner measurements corrupted with an additive noise:

$$y_t = y_0 + \int_0^t x_s(\theta)ds + \int_0^t G_s(Y_s,\theta)dw_s,\ 0 \le t \le T, \tag{1.1}$$

w is a Wiener process, $Y_t = \{y_s,\ 0 \le s \le t\}$, and the following are satisfied:

$$P(\int_0^T |x_t|dt < \infty) = 1\ ,$$

$$P(\int_0^T |G_t|dt < \infty) = 1,$$

where G_t is $Y_t = \sigma-\text{alg}\ \{Y_t\}$ measurable (non-anticipative) functional, and θ is the parameters vector.

Theorem 1 [8]: If $G_t G_t^*$ is uniformly nonsingular (i.e. the elements of the inverse matrix are uniformly (in t) bounded), and

$$\int_0^T E\{|x_t|^2\}dt < \infty,$$

then

$$\nu_t(\theta) = \int_0^t (G_s G_s^*)^{-1/2}(dy_s - E_\theta\ \{\ x_s|\ Y_s\}ds)\ , \tag{1.2}$$

is a (standard) Wiener process with respect to Y_t, i.e.

$$E_\theta\{\exp\,(iz^*(\nu_t(\theta) - \nu_s(\theta)))|\,Y_s\} = \exp(-0.5zz^*(t-s)),$$

and

$$dy_t = E_\theta\{x_t|\,Y_t\}dt + G_t d\nu_t(\theta)\ . \tag{1.3}$$

In the above and in the sequel * designates matrix transpose.

In order to use some of the results from the asymptotic estimation theory [5,9,10], the following parameter change model is postulated:

there exists $t_0 \in (0,\infty)$ such that

$$\theta = \begin{cases} \theta_0 & t < t_0\ , \\ \theta_0 + t^{-1/2}\delta\theta & t \geq t_0\ . \end{cases} \tag{1.4}$$

According to Theorem 1 we have

$$d\nu_t(\theta_0) = z_t\ 1(t - t_0)dt + dv_t\ , \tag{1.5}$$

where $1(t)$ is the unit step function, v is a Wiener process adaptable to Y_t and z_t is an unknown process which represents the error caused by using θ_0 rather than θ in the calculation of ν.

Let τ be a Markov time with respect to Y_t. τ is interpreted as the alarm, based on monitoring the process ν, to signal a change in system parameters. It is desired to choose τ as close to t_0 as possible. The risk (cost) associated with the choice of τ should consider:

$$P(\tau < t_0) = \text{probability of false alarm} \tag{1.6}$$

and

$$E\{\max(0,\tau - t_0)\} = \text{average alarm delay}\ .$$

The formal design of the stopping rule under such criteria leads to numerous difficulties and in practice requires seldomly-verifiable assumptions about the distribution of the change time instant [11]. In order to characterize the desired detection scheme we use the following modification of the result presented in [8].

Let $S(\tau)$ denote a sequential scheme of testing the following alternative hypotheses:

$$H_0 : d\nu = dw_t, \tag{1.7}$$

$$H_1 : d\nu_t = x_t dt + dw_t,$$

where w is a Wiener process independent of x_t. The scheme S is characterized by a Markov time $\tau(\nu)$ which corresponds to the time instant of accepting H_1. Let P_i and E_i denote the probability distribution and the expected value for the case when the H_i hypothesis is satisfied, $i = 0,1$. For each scheme S denote

$$e_1(S) = P_1(\tau = \infty) \text{ the error of the first kind }, \tag{1.8}$$

$$e_2(S) = P_0(\tau < \infty) \text{ the error of the second kind (false alarm)}$$

Consider the class S of detection schemes such that $e_1(S) \leq \alpha$ and $e_2(S) \leq \beta$, $\alpha + \beta < 1$.

Theorem 2 [8]:

Let

$$E_i \{ \int_0^T |m_t|^2 dt \} < \infty , i = 0,1 ,$$

where

$$m_t = E_1\{x_t | \sigma\text{-alg}\{\nu_s, 0 \leq s \leq t\}\} .$$

Then there exists a scheme $S_{opt}(\tau_{opt})$ defined below and optimal in the sense that for any other scheme $S(\tau)$ we have

$$E_i \{ \int_0^{\tau_{opt}} |m_t|^2 dt \} \leq E_i \{ \int_0^{\tau} |m_t|^2 dt \} . \tag{1.9}$$

The scheme S_{opt} is given by

$$\tau_{opt} = \inf \{t : \lambda_t \geq \lambda_{\max}\} , \tag{1.10}$$

where

$$\lambda_t = \sup_{p \leq t} \{ \int_p^t (m_s d\nu_s - 0.5|m_s|^2 ds) \} , \tag{1.11}$$

and $\lambda_{\max} = \ln((1 - \alpha)/\beta)$.

Comments:

The basic conclusion from the above theorem is that the log-likelihood ratio testing scheme is optimal in the sense of (1.9). This result extends the Neyman-Pierson test providing the "shortest-weighted" detection delay, for fixed probabilities of the first and second type errors.

The here postulated independence of w and x is not generally satisfied for the above proposed parameter change model. The sup operation in (1.11) is performed over the functional which can not be evaluated directly since m is not available to the user. To gain some insight into the properties of (1.11) assume that $m_t = m = const$, and that the conditional distribution reaches its maximum at m. Replacing m by its most likely value under hypothesis H_1 we have:

$$\lambda_t = 0.5 \sup_{p \le t} \{(t-p)^{-1} \|\nu_t - \nu_p\|^2\} \ . \tag{1.12}$$

From the above it is seen that even in this oversimplified case the thresholding test (1.10) is of a complex nature since only the argument of sup operation is simply related to the observations of ν.

The threshold $\lambda_{\max}$ used in the above theorem must be viewed as a purely conceptual one. A discussion on selecting an appropriate threshold is given for the discrete-time case. Here we note that even in a simple case the threshold selection is a complicated process. As an example consider the scalar version of (1.12). Denote by w_t a standard Wiener process. Then for fixed p and under the hypothesis H_0 we have:

$$P\{(t-p)^{-1} \, |\nu_t - \nu_p|^2 \le \epsilon(t-p) \ , t-p > \delta\} =$$

$$P\{|w_t| \le t\epsilon^{1/2}, \ t > \delta\} =$$

$$P\{|w_{1/t}| \le t\epsilon^{1/2}, \ t > \delta\} =$$

$$P(|w_t| \le \epsilon^{1/2}, \ 0 < t < \delta^{-1}\} =$$

$$P\{|w_t| \le \gamma, \ 0 < t < 1\} =$$

$$P\{\sup_{0<t<1}|w_t|\leq\gamma\}=$$

$$\sum_{-\infty}^{\infty}(-1)^k(2\pi)^{-1/2}\int_{-\gamma}^{\gamma}\exp(-0.5z+k\gamma)dz\,, \tag{1.13}$$

where

$$\gamma=(\delta\epsilon)^{1/2}\,.$$

In the above the fact that if w_t is a Wiener process so are its transformations : $(tw_{1/t},t>0$, and $(s^{1/2}w_{t/s},s>0)$ are used. The last equality follows from the reference [12]. Now, for each t it is possible to estimate the probabilities of crossing a specified threshold $\lambda_{\max}$. We note here that equation (1.13) can be used to derive tables of thresholds, corresponding to different probabilities of the false alarm.

To justify further the model for parameter changes used in the above derivation consider a special version of (1.1):

$$dy_t=(h(t,Y_t)+H(t,Y_t)\theta)dt+G(t,Y_t)dw_t\,. \tag{1.14}$$

If the parameter θ is unknown (e.g. after abrupt change) and we model the uncertainty about θ by interpreting it as a random variable with mean θ_0 and certain distribution $P(\theta)$ we can use a conditionally linear filter [13] to derive the following formula for estimation of unknown θ:

$$d\hat{\theta}_t=K_t(dy_t-(h(t,Y_t)+H(t,Y_t)\hat{\theta}_t dt)\,,\ \hat{\theta}_0=\theta_0\,, \tag{1.15}$$

where

$$K_t=(G(t,Y_t)G(t,Y_t)^*)^{-1}P_tH^*,$$

$$P_t=cov(\hat{\theta}_t|\,Y_t)\,.$$

Under somewhat technical assumptions the following convergence rate of $\hat{\theta}$ to θ can be proved [8]:

$$E_\theta\{inf\,\{t:|\hat{\theta}_s-\theta\,|\leq\epsilon,s\geq t\}\}=const\;\epsilon^{-2}\,. \tag{1.16}$$

Equation (1.16) can be interpreted as convergence with the average rate not worse than $t^{-1/2}$. Thus if the detection algorithm is associated with the parameter estimation algorithm a "bootstrap" behavior of both of these algorithms can be observed, (e.g. the

detection alarm may trigger re-initialization of the parameter estimator gains and the convergence of the estimator to the true parameter value may follow the underlying parameter change model with θ_0 replaced by $\hat{\theta}$).

2. Discrete Time Detection

The results in this section are based on the work published in [3] and [14]. The new results presented here for the first time include robustness of sequential testing and application of the fluctuation theory to threshold selection.

Assume first that the observable data satisfy:

$$y_t = f(x_t\ ,\ x_{t-1},\cdots,x_{t-n}),\ t \geq 0,\ \ n \geq 0\ , \tag{2.1}$$

where $\{x_t\}$ is a Markov chain such that

$$dP(x_t|x_{t-1},\cdots,x_1) = dP(x_t|x_{t-1};\theta)\ , \tag{2.2}$$

with $P(\cdot|\cdot)$ being the transition probability of a Markov chain $\{x_t\}$, and θ a piecewise constant "parameter" function.

The *Change Detection Problem* in its original form is stated as follows:
there exists an instant $t_0 \in (0,\infty)$ such that

$$\theta = \theta_0 \text{ for } t < t_0 \tag{2.3}$$

$$\theta = \theta_1 \text{ for } t \geq t_0$$

At any time t we want to decide between two hypothesis:

$$H_0{:}t{>}t_0 \text{ (no parameter change occurred before } t)\ , \tag{2.4}$$

$$H_1{:}t{\leq}t_0 \text{ (a parameter change occurred prior to } t).$$

Once H_1 is accepted we want to estimate the change time t_0.

Assume that we are using the following recursive procedure for parameter estimation:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t H(\hat{\theta}_{t-1},y_t)\ . \tag{2.5}$$

Assume also that $\{x_t\}$ is asymptotically ergodic and that

$$h(\hat{\theta},\theta) = \lim_{t\to\infty} E_{\hat{\theta},\theta}(H(\hat{\theta},y_t)) = 0 \Longleftrightarrow \hat{\theta} = \theta\ , \tag{2.6}$$

where $\hat{\theta}$ is the "adjustable" parameter (estimator) and θ is the "true" parameter (not available to the user).

Comments:

- the class of algorithms (2.5) includes in particular the least-squares stochastic gradient algorithm:

$$H(\hat{\theta},y_t) = (df(x_{t-1};\theta)/d\theta)^*(x_t - f(x_{t-1};\theta)) \mid \theta = \hat{\theta}\ , \tag{2.7}$$

  where $x_t = f(x_{t-1};\theta) + w_t$, and $\{w_t\}$ is a white-noise sequence, and both ergodicity and stability are taken in the same sense.

- the asymptotic ergodicity holds for example, for stable conditionally linear systems of the following form:

$$y_t = F(\hat{\theta}_{t-1})y_{t-1} + G(\hat{\theta}_{t-1})w_t\ , \tag{2.8}$$

  where $\{w_t\}$ is a white-noise sequence, and both ergodicity and stability are taken in the same sense.

- in (2.6) $E_{\hat{\theta},\theta}$ denotes the expectation with respect to the steady-state distribution of $\{y_t\}$. Assumption (2.6) is difficult to verify, however it is a standard assumption in recursive estimation schemes.

In order to apply the asymptotic local approach of LeCam theory [5,9,10,15] to construct a sequential detection algorithm, we modify the *Change Detection Problem* as follows:

Given a "nominal model" $\theta = \theta_0$, and a data record $(y_0,\ldots,y_t)$, of length t. Using the random vector $H(\theta_0;y_k)$, $0 \le k \le t$, we want to test sequentially between the following hypothesis:

$$H_0 : \theta = \theta_0 \tag{2.9}$$

$$H_1 : \text{there exists } t_0 \in (1,t] \text{ such that:}$$

$$\theta = \theta_0 \text{ for } k < t_0$$

$$\theta = \theta_0 + \delta\theta k^{-1/2} \text{for } t_0 \le k \le t\ ,$$

where $\delta\theta \neq 0$ is an unknown change.

Theorem 3 [3]:

Let

$$\varphi_{t,s} = t^{-1/2} \sum_{k=1}^{\lfloor st \rfloor} (H(\theta_0; y_k)) \ , \ 0 \leq s \leq 1 \ . \tag{2.10}$$

Then $\varphi_{t,s}$ converges (weakly) *to* v_s as $t \to \infty$, where

$$dv_s = \alpha \ 1 \ (st \geq t_0) ds + R^{1/2}(\theta_0) dw_s \ , \tag{2.11}$$

w_s is a Wiener process, $\alpha \neq 0$, and

$$R(\theta) = \lim_{t \to \infty} \sum_{k=1}^{t} (cov_{\theta,\theta})(H(\theta; y_k), H(\theta; y_1)) \ . \tag{2.12}$$

Conclusion:

$$\varphi_{t,1} \ t \to \infty \begin{cases} N(0, R(\theta_0)) \text{ under } H_0 \\ N(\alpha, R(\theta_0)) \text{ under } H_1 \end{cases} . \tag{2.13}$$

$N(.,.)$ denotes here normal distribution.

Theorem 3 has the following practical implications:

assuming that a change occurred at time t_0 and that t is sufficiently large, the random vectors $H(\theta_0, y_k)$ are independent and distributed as follows:

$$H(\theta_0, y_k) \sim N(0, R(\theta_0)), k < t_0, \tag{2.14}$$

$$H(\theta_0, y_k) \sim N(\alpha, R(\theta_0)), \ k \geq t_0 \ .$$

Hence the parameter change detection problem is replaced by the asymptotically equivalent problem of detecting changes in the mean of independent Gaussian random variables.

Change Detection Algorithm:

Let

$$V_t = H(\theta_0; y_t) \ . \tag{2.15}$$

For the record $\{V_1, \ldots, V_t\}$, the log-likelihood ratio between:

H_0: the mean of V_t is equal to 0,

and

H_1: starting from $t_0 \leq t$ the mean of V_t equals α,

is given by

$$S(t,t_0,\alpha) = \sum_{k=t_0}^{t} (\langle V_k, R^{-1} V_k \rangle - \langle V_k - \alpha, R^{-1}(V_k - \alpha) \rangle) \tag{2.16}$$

replacing α by its most likely value under H_1 we get

$$S(t,t_0) = \max_{\alpha}\{S(t,t_0,\alpha)\} = (t-t_0+1)^{-1} \langle \sum_{k=t_0}^{t} (V_k),\ R^{-1} \sum_{k=t_0}^{t} (V_k) \rangle \ .$$

In order to estimate t_0 we maximize $S(t,t_0)$ with respect to t_0:

$$t_0(t) = \operatorname{argmax}_{t_0}\{S(t,t_0)\} \ .$$

Denote:

$$\lambda_t = S(t,t_0) = \max_{t_0}\{S(t,t_0)\} \ . \tag{2.17}$$

The detection time of a parameter change is now given by the stopping time t_d:

$$t_d = t_0(\min\{t{:}\lambda_t \geq \lambda_{\max}\}) \ ,$$

where $\lambda_{\max}$ is a given threshold which controls the probabilities of false change detection and detection delay.

Note:

λ_t does not have χ^2 distribution. The threshold $\lambda_{\max}$ must be calculated using the distribution of the maximum of a sum of independent Gaussian variables.

Let $\{x_i\}$ be i.i.d., and $S_n = x_1 + x_2 + \cdots + x_n$, $S_0 = 0$.

The Fluctuation Theory studies random variables of the form $f(S_0,S_1,\ldots,S_n)$. Here the important case is

$$M_n = \max\{0,S_1,\ldots,S_n\} \ ,$$

and its modification

$$N_n = \max\{0,S_1,2^{-1/2}S_2,\ldots,n^{-1/2}S_n\}.$$

It can be shown that for $|p| < 1$ and any real u [16,17,18]:

$$\sum_{n=0}^{\infty} E\{\exp(iu\ M_n)\}p^n = \exp(\sum_{k=1}^{\infty}(p^k/k)E\{\exp(iu\ \max\{S_k,0\})\}\ . \qquad (2.18)$$

Equation (2.18) can be used to derive various statistical parameters of M_n. In particular:

$$E\{M_n\} = \sum_{k=1}^{n} k^{-1}E\{\max(S_k,0)\}$$

If $x_i \sim N(0,1)$ then $S_k \sim N(0,k)$ and

$$E\{M_n\} = (2\pi)^{-1/2}\sum_{k=1}^{n}(k^{1/2})\ ,$$

and

$$E\{M_n{}^2\} = 0.5n + (2\pi)^{-1}\sum_{k=1}^{n-1}((k^{-1/2})\sum_{i=1}^{n-k}(i^{-1/2}))\ .$$

Also if $P(S_k \geq 0) = 0.5$ then [16]

$$P(\text{first maximum occurred for } S_k) = 2^{-2n}\binom{2k}{k}\binom{2(n-k)}{n-k}\ .$$

Using these relationships we can estimate $E\{N_n\}$ from the above and accordingly set the threshold $\lambda_{\max}$ (see the numerical example). Further research on better N_n distribution approximation is necessary. Some of the useful approximations can be derived using the Donsker's Theorem [10].

Remarks:

The assumptions about the properties of the random field $H(\ )$ are in general difficult to verify, and the corresponding results are valid only asymptotically. This exemplifies the importance of using here a robust algorithm. The asymptotic approach provides merely the guidelines for the detection algorithm design.

The results of fluctuation theory aid construction of so-called Huber estimators [19,20,21], whose goal is to achieve robustness of the otherwise non-robust log-likelihood method. In the numerical example presented in this paper the following robust modification of each V_k is used:

$$V_{\text{mod}_k} = \max\{-a(n),\min\{a(n),V_k\}\},\ \ n = \text{sample size}\ ,$$

where $a(n)$ and the thresholds (as a function of k) are obtained, for a fixed probability of

the false alarm, from (2.18). V_{mod} is used in the detection formulas (2.16-17) instead of V.

It should be emphasized that the application of the robust statistics here is warranted by the fact that the statistical test is reduced to one of its simplest, standard forms.

Example:

$$x_t = f(x_{t-1};\theta) + g(\theta)w_t,$$

where $\{w_t\}$ is a standard white noise sequence.

Define $y_t = [x_t, x_{t-1}]$.

Then

$$H(\theta,y_t) = (df(x_{t-1};\theta)/d\theta)^*(x_t - f(x_{t-1};\theta)) ,$$

when a stochastic gradient method for parameter estimation is used.

Accordingly

$$R(\theta) = g^2(\theta)E_\theta((df(x_0;\theta)/d\theta)^*(df(x_0;\theta)/d\theta) .$$

If the model is linear-in-parameters i.e.

$$f(x_t;\theta) = F(x_t)\theta$$

then

$$R(\theta) = g^2(\theta)E_\theta(F^*(x_0)F(x_0)) .$$

Implementation Note:

There is a need to store an increasing sequence of scalar values $S(t,t_0)$ and to find its maximal element. This should not be a concern for most of the biological "real-time" experiments using even slow (PC-type) computers and long data records. If high performance is required it should be noted that the operation of finding a maximum in the detection formulae can be implemented in a systolic type architecture.

3. Numerical Example

To illustrate some of the implementation aspects of the proposed detection algorithm a simple 2-nd order dynamical system of so-called Lotka-Volterra type is simulated. Such a model does not attempt to describe any of the immunological functions in detail, however it has a characteristic qualitative behavior of competing agents.

$$dx(t)/dt = \theta^{(1)}x(t) - \theta^{(2)}x(t)y(t) \,, \tag{3.1}$$

$$dy(t)/dt = -\theta^{(3)}y(t) - 1) + \theta^{(4)}x(t)y(t) \,,$$

where $\theta^{(i)} \geq 0$, $i = 1,2,3,4$, are the parameters.

In its very crude form $x(t)$ may represent the concentration of cancer cells or a virus and $y(t)$ characterizes the immune system (e.g. T-lymphocyte concentration). It is assumed that only $y(t)$ is observed and that the corresponding measurements are made at the discrete time instants with a random error:

$$y_k = y(t_k) + Qw_k \,, \ \{w_k\} = \text{standard white noise} \,. \tag{3.2}$$

Formally a continuous-discrete time filter should be used here. Instead much simpler, but also less accurate, discrete time filter is applied. The corresponding equations are given below (see also Appendix A):

First the discretization of (3.1) which includes measurement error takes the following form:

$$x_{t+1} = F(y_t)x_t + G(x_t,y_t)w_{t+1} \,,$$

$$y_{t+1} = h(y_t) + H(y_t)x_t + R(x_t,y_t)w_{t+1} + Qv_{t+1} \,,$$

where

$$F(y_t) = \exp(\delta(\theta_0^{(1)} - \theta_0^{(2)}y_t)) \,,$$

$$G(x_t,y_t) = Q\delta\theta_0^{(2)}\, F(y_t)x_t \,,$$

$$H(y_t) = (1 - \exp(-\delta\theta_0^{(3)}))\theta_0^{(4)}\, y_t(\theta_0^{(3)})^{-1} \,,$$

$$h(y_t) = (y_t - 1)\exp(-\delta\theta_0^{(3)}) + 1 \,,$$

$$R(x_t,y_t) = - Q(\exp(-\delta\theta_0{}^{(3)}) + (1 - \exp(-\delta\theta_0{}^{(3)})\theta_0{}^{(4)}x_t(\theta_0^{(3)})^{-1} .$$

The next step of approximation consists of replacing x_t in $G()$ and $R()$ by its best mean-square estimator and applying conditionally linear filter equations, which yields

$$m_{t+1} = F(y_t)m_t + K(y_t)B(y_t)(y_{t+1} - h(y_t) - H(y_t)m_t) ,$$

$$B(y_t) = (Q^2 + R^2(y_t) + H^2(y_t)P_t)^{-1/2} ,$$

$$K(y_t) = (G(m_t,y_t)R(m_t,y_t)+F(y_t)H(y_t)P_t)B(y_t) ,$$

$$P_{t+1} = F^2(y_t)P_t + G_2(m_t,y_t) - K^2(y_t) .$$

We note that the above approximation being of a crude nature is sufficient for the purpose of change detection. This again emphasizes the robustness of the algorithms used.

In addition to the robust modification of the log-likelihood test a heuristic smoothing of the alarm function was used. A simple averaging of the alarm function within a fixed length window allows an increase in the range of measurement variances without causing the false alarms. The averaging window length increases in an obvious fashion the detection delay. For the sake of graphical data presentation the measurement points are generated prior to invoking the sequential test algorithm.

All the calculations were performed using the PC-MATLAB ver.3.13 program on an IBM-PC. The listings of written subroutines are presented in Appendix B. The description of commands and standard subroutines can be found in the PC-MATLAB User's Manual.

The Figures 1 to 12 represent graphical output of several numerical experiments. The following parameter values were used:

Parameters before the change:

$\theta^{(1)} = 1.00,\ \theta^{(2)} = 1.00 ,\ \theta^{(3)} = 1.00,\ \theta^{(4)} = 1.00 .$

Parameters after the change:

$\theta^{(1)} = 1.00,\ \theta^{(2)} = 0.40 ,\ \theta^{(3)} = 1.00,\ \theta^{(4)} = 0.20 .$

The change time = 2, final time = 3.

Measurement error variance and sampling period is listed for each run separately. The change detection status is given above the continuous time trajectories graphs.

Figure 1 and 2 show measurement data and system response respectively with no change in parameters.

Figures 3, 4, 5 and 6 show the detection results for the same measurement error variance but different sampling periods.

Figures 7 and 8 show improvement in detection performance due to the decreased measurement error variance.

Figures 9, 10, 11 and 12 show the effect on detection performance of increasing the sampling period. In the case shown on Figures 11 and 12 the detection threshold was increased to avoid false alarms. This adjustment was necessary due to the increased systematic error of discrete-time model approximation.

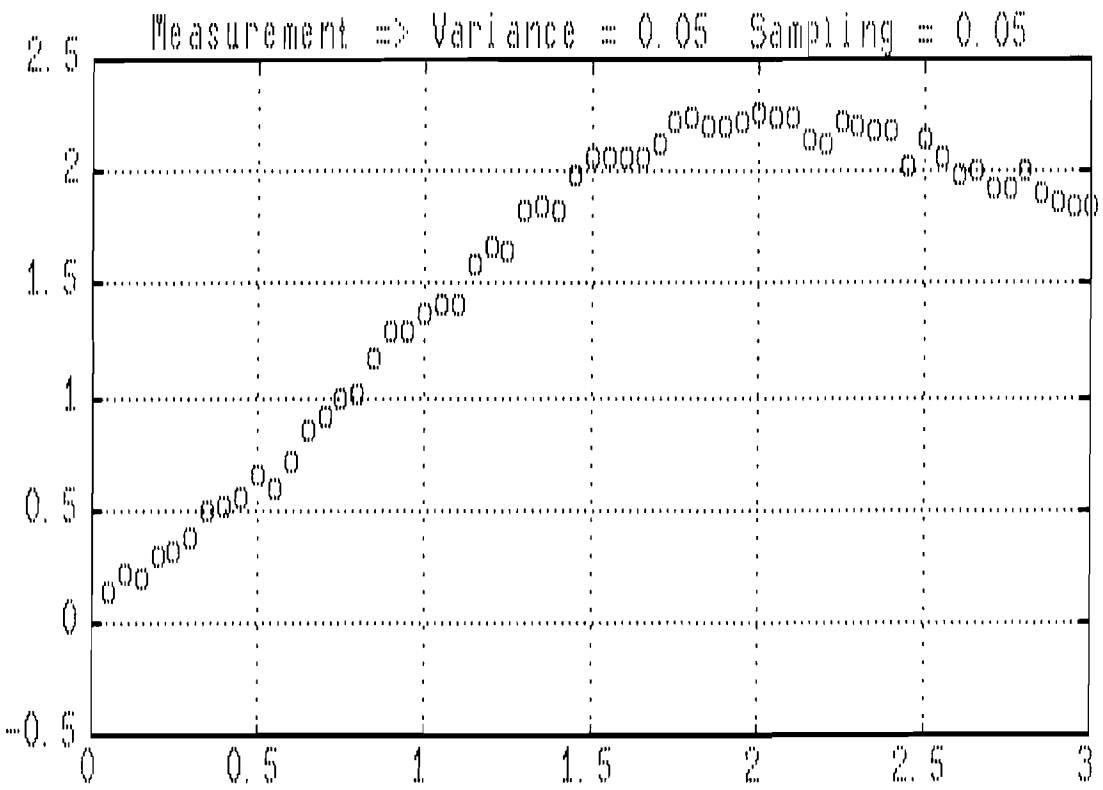

Figure 1

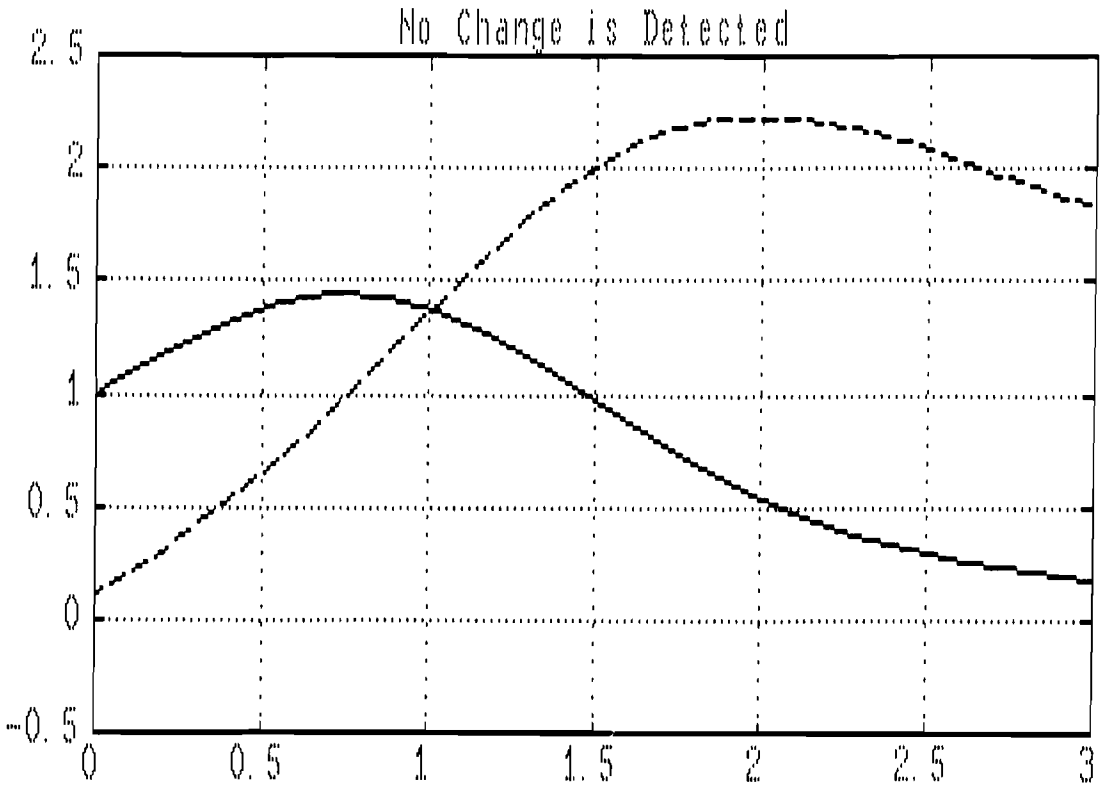

Figure 2

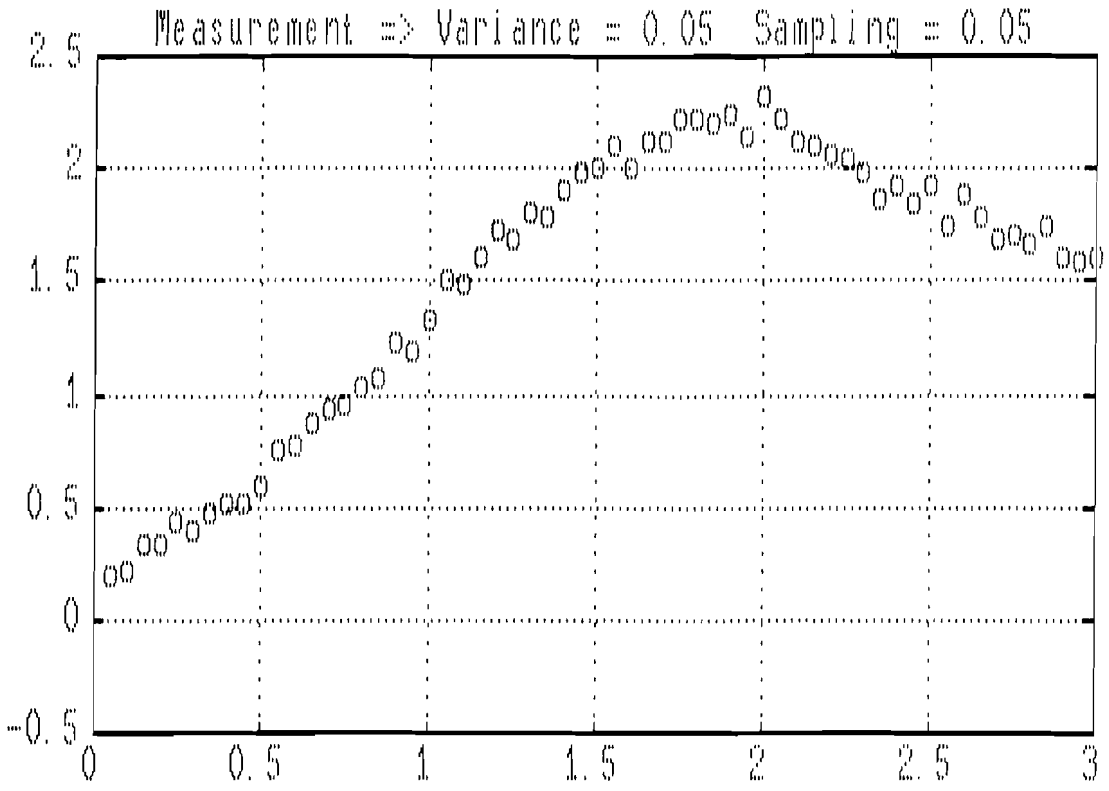

Figure 3

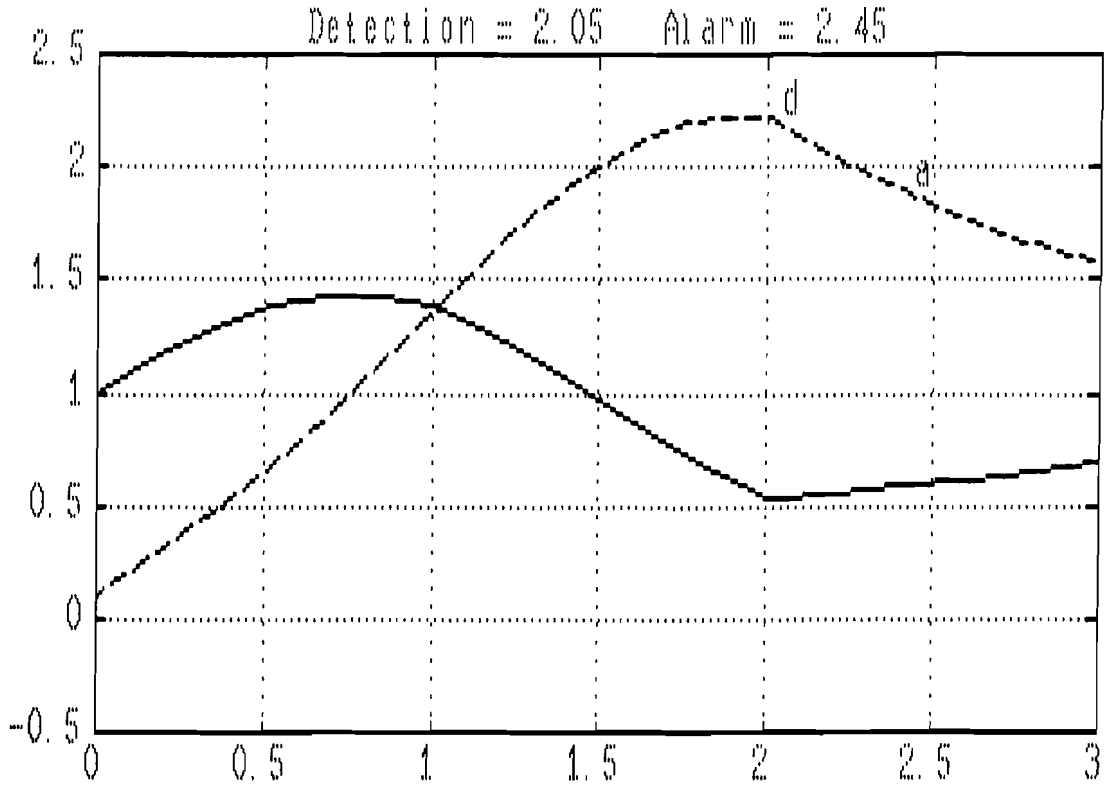

Figure 4

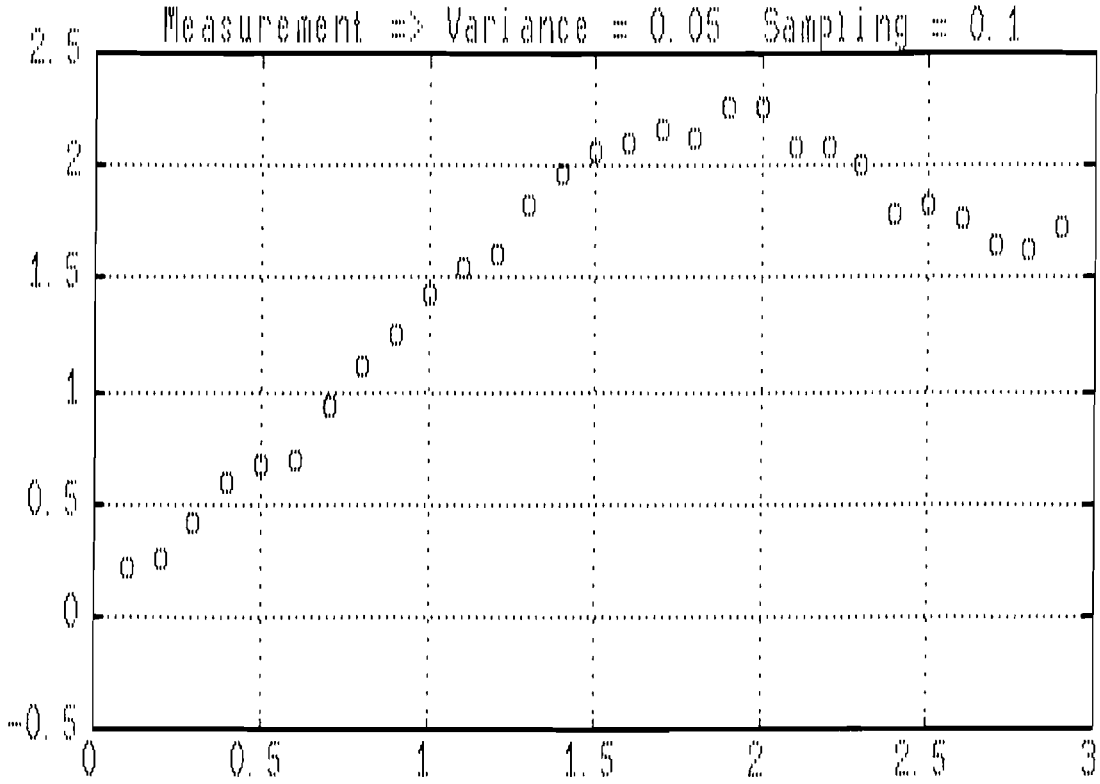

Figure 5

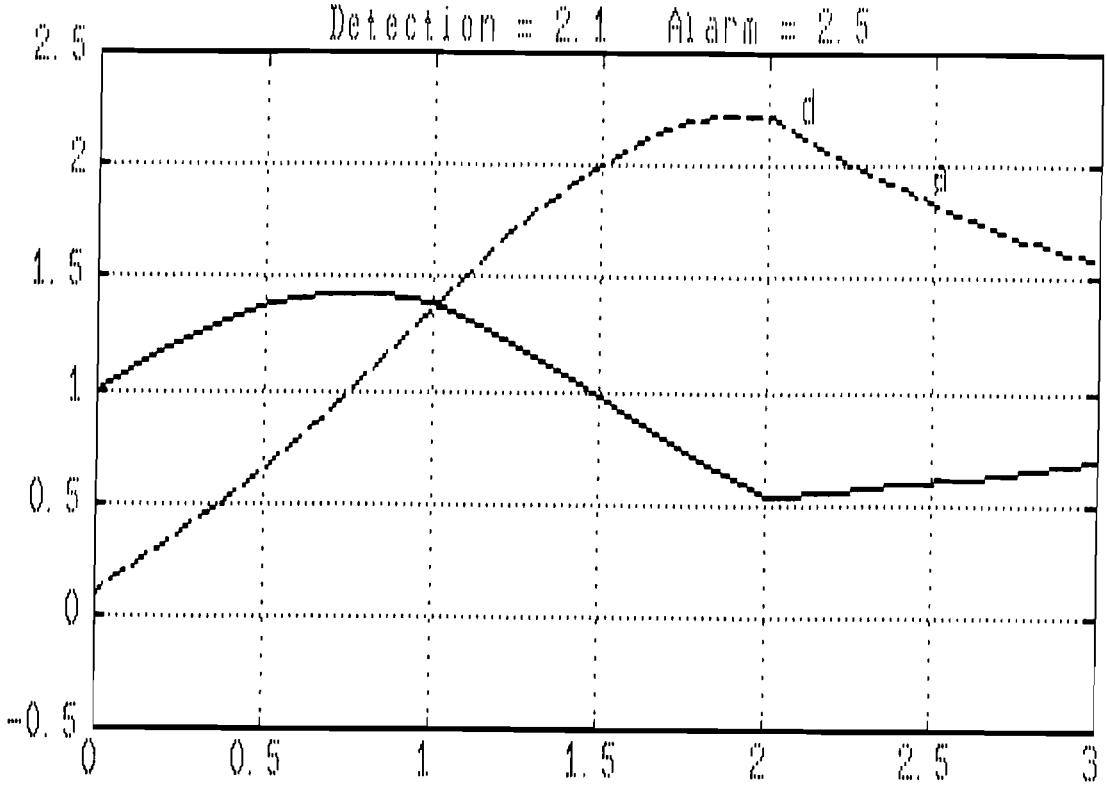

Figure 6

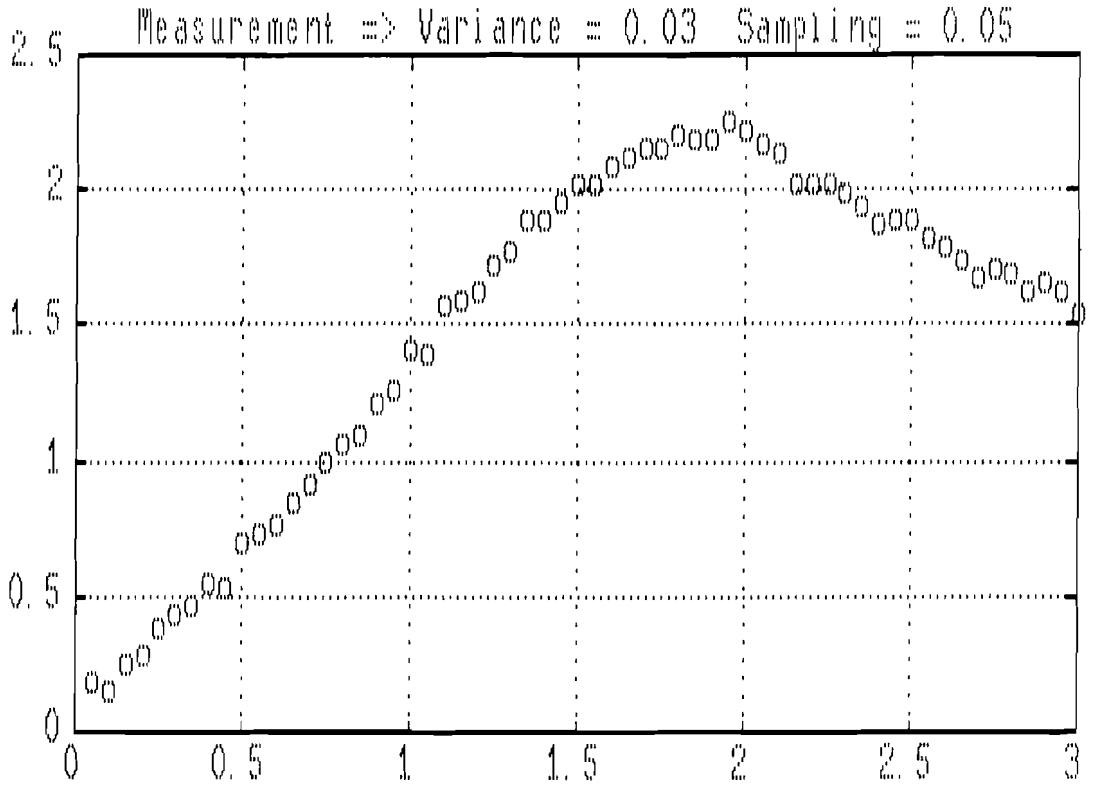

Figure 7

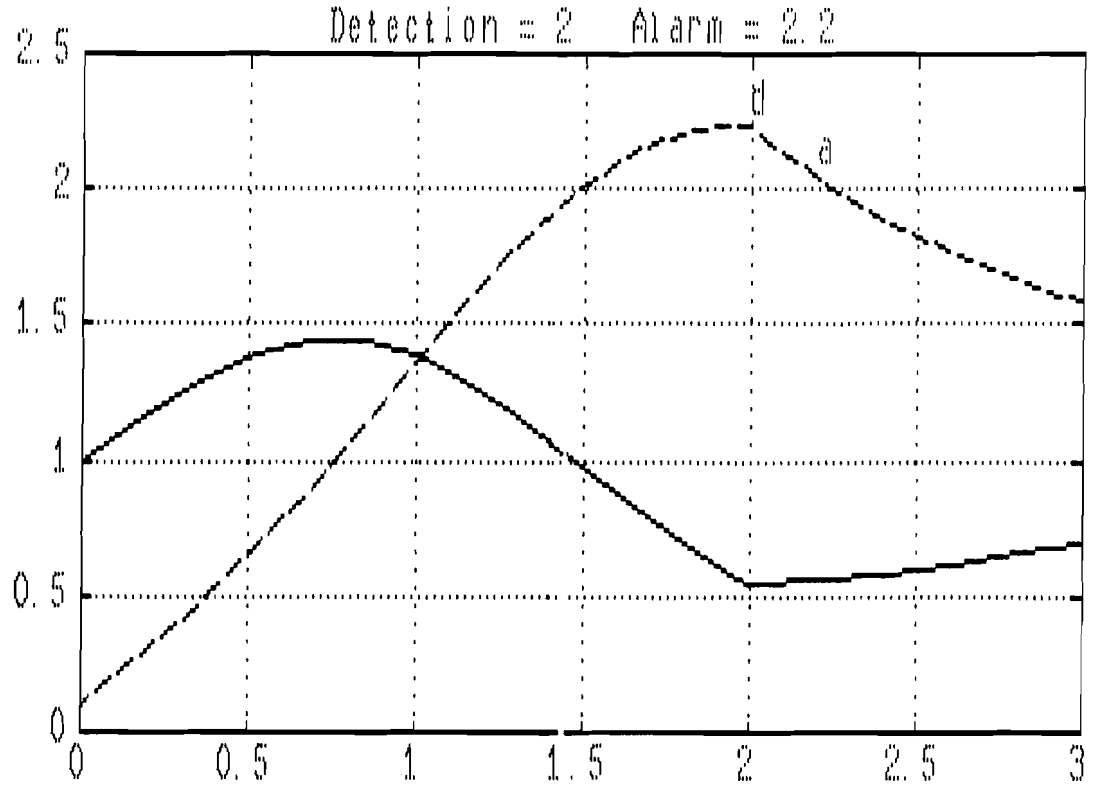

Figure 8

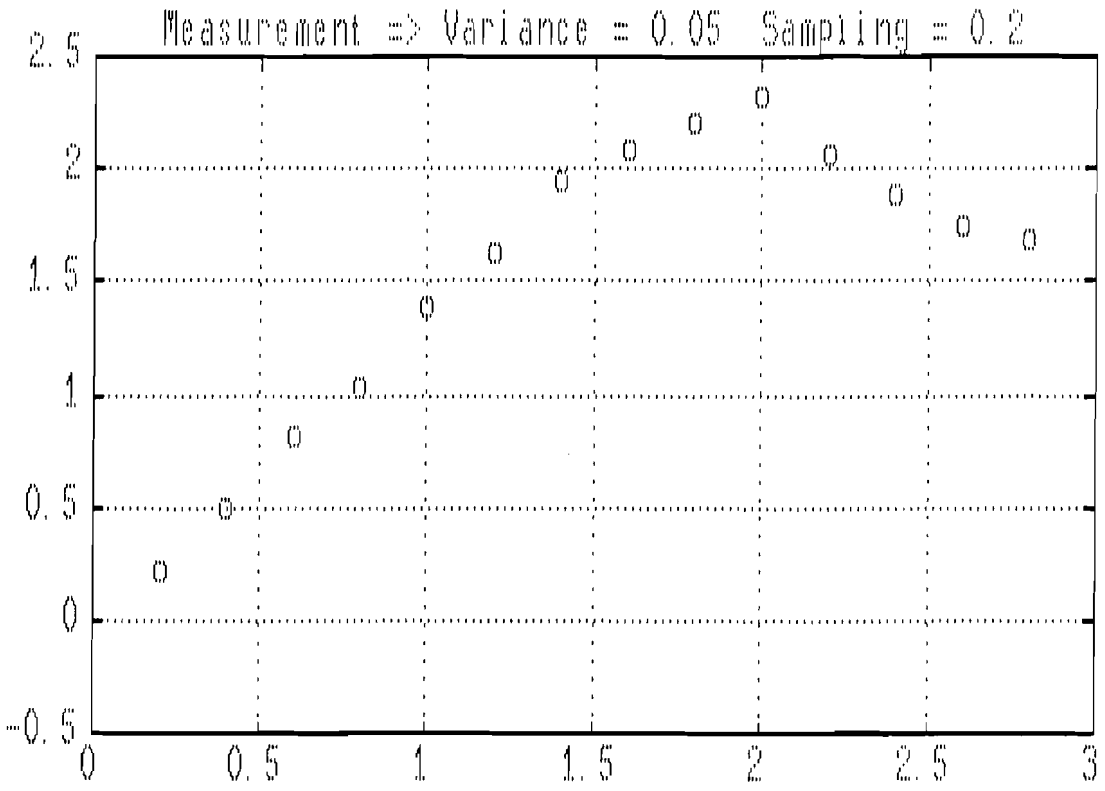

Figure 9

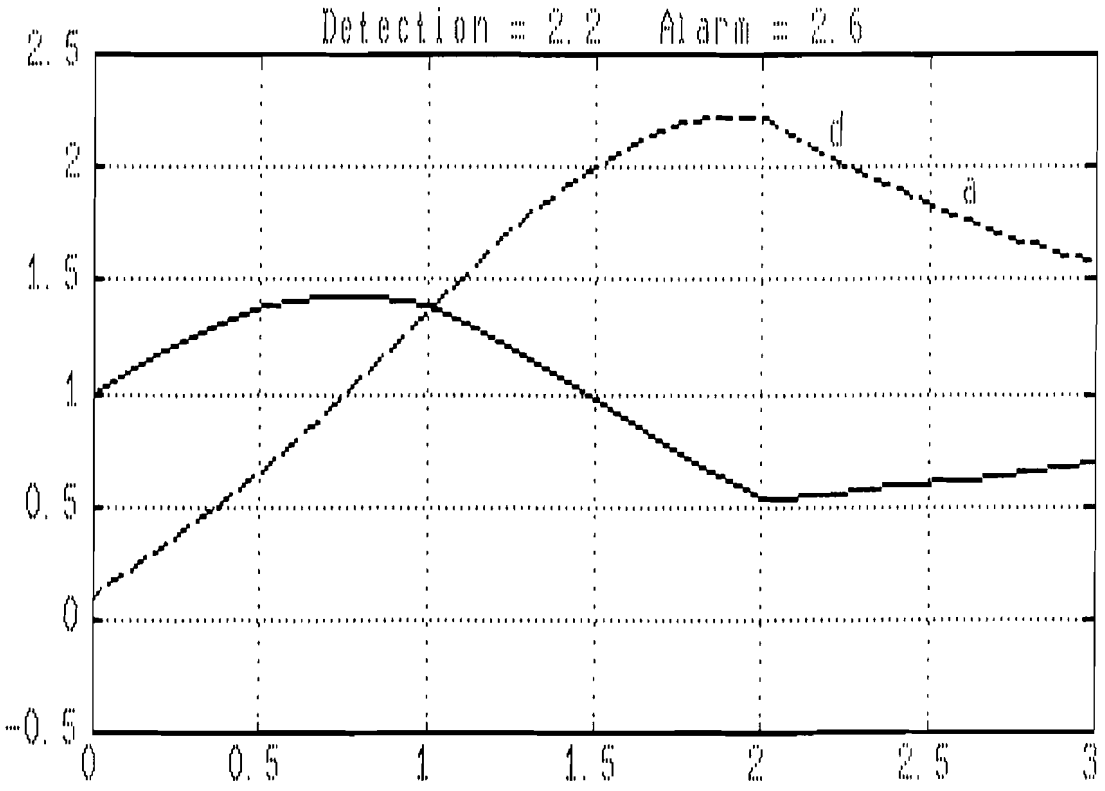

Figure 10

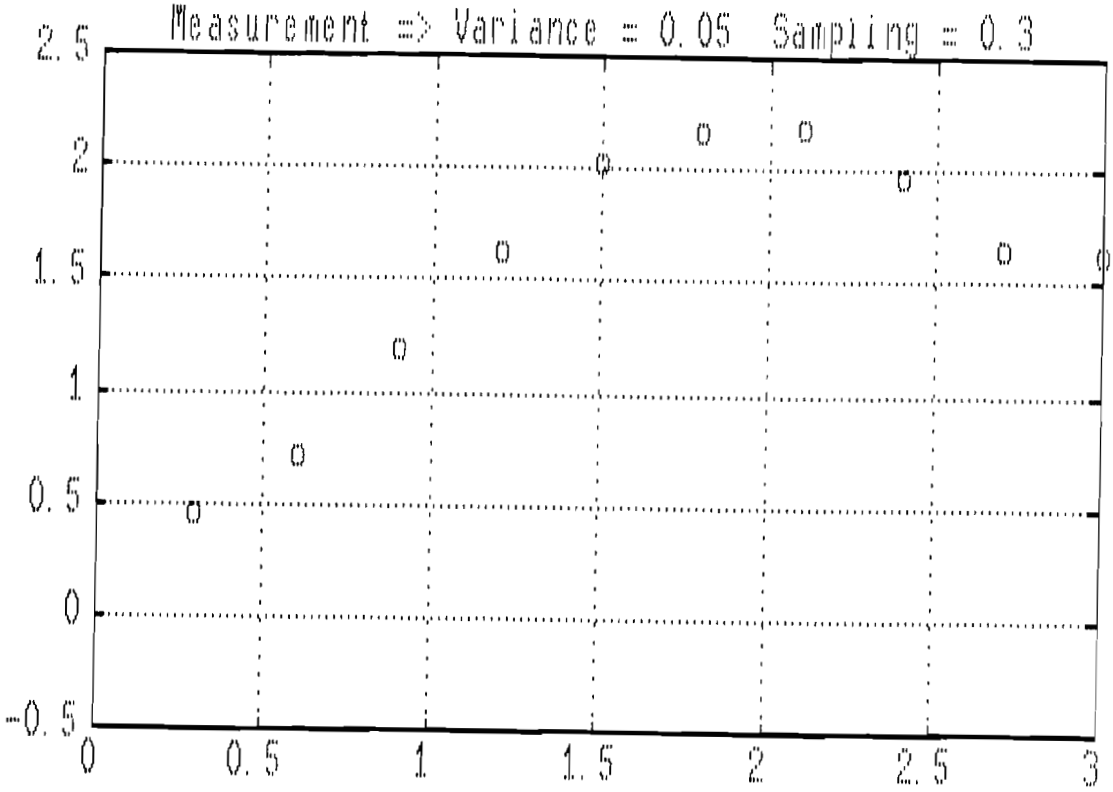

Figure 11

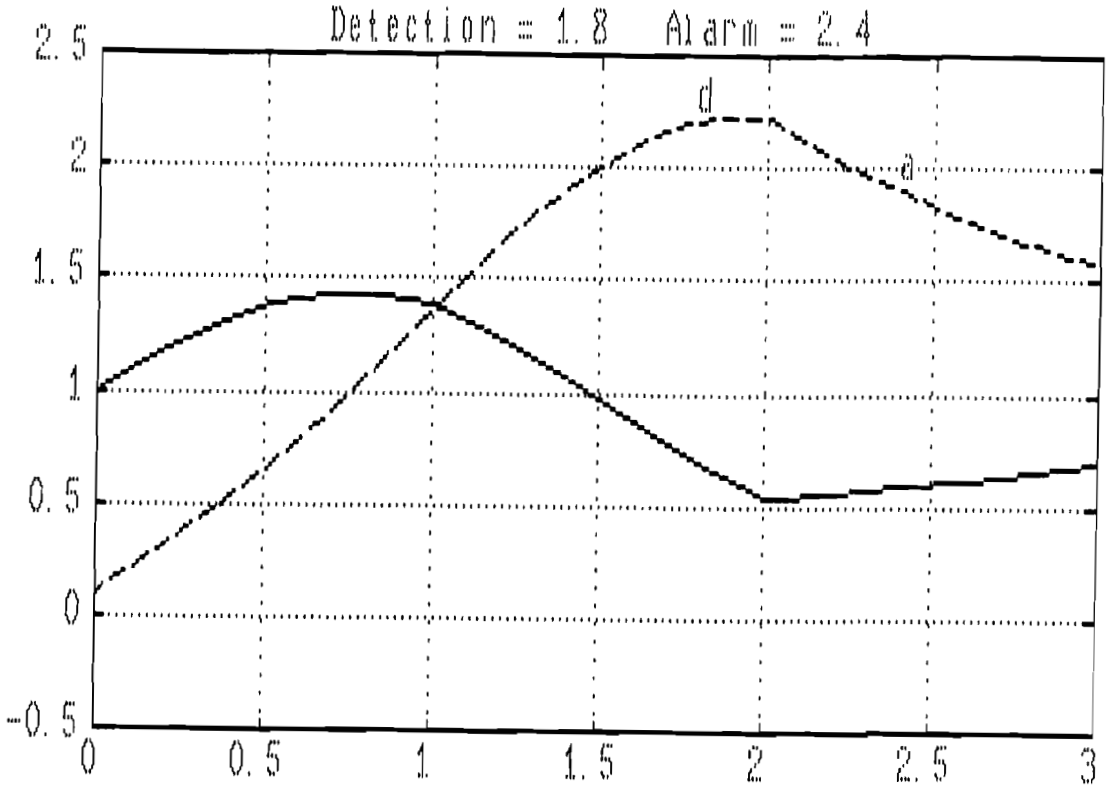

Figure 12

Conclusions

The statistical approach to data processing and sequential testing has a long tradition. Probably the most successful example of such an approach is the heuristic work of Wald [22]. The main criticism of statistical methods comes from the fact that some of the experiments do not provide sufficient data samples to verify the statistical assumptions used in model derivation. One may argue that the immunological data belongs to the latter category. Here the statistical testing is proposed to implement a parameter change algorithm. The main justification in using this approach is the need of having a uniform theoretical framework to design, test and compare various detection methods under different experimental conditions. Since the basic concepts of statistics are known and accepted in practice by most engineers and scientists it seems natural to use this as a base while proposing new algorithms and techniques. Knowing that the postulated mathematical properties of the modeled signals and systems are not easily verifiable should emphasize robustness of derived algorithms. Again the field of robust statistics is gaining recognition in the last few years and maturing to the point of being a useful engineering tool.

The detection algorithms studied in this paper are based on sequential hypotheses testing in a so-called local asymptotic framework. The ability of reducing a complex in nature test to a simple test for a non-zero mean of a sequence of random variables is an attractive practical result. The simple form of such a test allows further experimentation with robust modifications as seen in the presented examples. It is apparent that the real performance of the derived methods can be verified only in a numerical manner. Here a simple numerical example is provided to highlight some of the concepts and to provide a basis for further investigation. Despite its simplicity this example may have practical applications in clinical oncology. An actual problem of clinical oncology is the prediction of individual reaction of a tumor process on the method of treatment. A characteristic behavior of the clinical data for the patients with stomach cancer takes the form of so-called breaking trajectories. Early detection of this behavior may assist medical treatment and aid in the prediction of the patients recovery process [7].

One of the conclusions resulting from the numerical experiment is that there is an obvious trade-off between the number of data points required for detection and the variance of measurement error. This could be of potential use to the experimentalists since it may allow trading between sparser sampling versus more accurate laboratory data processing. From the nature of biological data sources the sampling rates are relatively slow and often are barely capable of keeping up with the system's own dynamics. Therefore the assumption that the statistical averaging takes place is critical and must be verified for

each case individually. From the numerical cases presented it is relatively easy to see that the detection works in a non-trivial manner, providing differentiation between data records which seem to be graphically identical. Again this is where the theory meets practice and where the performance of the algorithm should be evaluated only on its application level. The examples presented here use the extended innovations concept. It should be emphasized that any random field which signals the difference between real parameters and their estimators can be used here. The work on the continuous detection method is currently in progress. Some of the continuous time results used in this paper are proved in [23].

Appendix A:

Conditionally linear filter equations are presented in [13], [14]. We give the corresponding result from the references. Consider a nonlinear time series $\{x_t\}$, $t \geq 0$, generated by the equations:

$$x_{t+1} = f + Fx_t + Gw_{t+1} + Qv_{t+1}\,, \tag{A.1}$$

$$y_{t+1} = h + Hx_t + v_{t+1}\,,$$

where f ,F , G ,Q ,h ,H are known functions of an argument $(t, Y_t = \{y_0, \ldots, y_t\})$.

The initial condition x_0 is a random variable for which the conditional distribution $P(x_0 \leq a | y_0) = P(a)$ is given ($x_0 \leq a$ denotes system of inequalities with respect to the corresponding components of x_0 and a). Sequences $\{w_t\}$, and $\{v_t\}$ are white, Gaussian, mutually independent and independent of x_0 and y_0. Note that the above model includes a class of nonstationary stochastic models which are nonlinear in the measured data and accept arbitrary distribution of the initial condition.

Let $p(x_t | Y_t)$ denote the conditional density of x_t given the observations Y_t. Then $p(x_t | Y_t)$ is given by

$$p(x_t | Y_t) = \int (g(x_t; m_t(a), P_t)\psi_t(a)\,dP(a)) \left(\int (\psi_t(a)\,dP(a))\right)^{-1}\,, \tag{A.2}$$

where $g(\cdot; m, P)$ denotes the Gaussian density with mean m and variance P

$$\psi_t(a) = \exp(-0.5 a^* S_t a + a^* R_t)\,,$$

and

$$m_t(a) = \varphi_t a + z_t\,,$$

$$z_{t+1} = f + Fz_t + K_t(y_{t+1} - h - Hz_t)\ ,\ z_0 = 0\ ,$$

$$K_t = (Q + FP_tH^*)(I + HP_tH^*)^{-1}\ ,$$

$$P_{t+1} = (FP_tF^* + GG^* + QQ^*) - K_t(Q + FP_tH^*)^*\ ,\ P_0 = 0\ , \qquad \text{(A.3)}$$

$$\varphi_{t+1} = (F - K_tH)\varphi_t\ ,\ \varphi_0 = I\ ,$$

$$S_{t+1} = S_t + \varphi_t^*H^*(I + HP_tH^*)^{-1}H\varphi_t\ ,\ S_0 = 0\ ,$$

$$R_{t+1} = R_t + \varphi_t^*H^*(I + HP_tH^*)^{-1}(y_{t+1} - h - Hz_t)\ ,\ R_0 = 0\ .$$

Note: Equation (A.1) can be evaluated explicitly in several special cases, e.g., when $P(a)$ is a mixture of Gaussian distributions.

Appendix B:

The command file and the subroutines used within PC-MATLAB environment to implement the detection algorithm are listed below. Some of the model parameters can be changed in an interactive fashion (i.e. differential equation coefficients, change time, terminal time, sampling period, measurement error variance). The other parameters can be changed by simply editing corresponding files. The file 'therdis.mat' contains pre-calculated alarm thresholds. This data file was created using the command file 'therdata.m'. The role of the flags and auxiliary variables is documented in the source files.

```
     ********************************************************
     *  Startup Procedure for Detection Algorithm Example   *
     ********************************************************

% Initialize Memory and Display
  clear;
  clear functions;
  hold off;
  clc;

% Parameters  Format (Example Values)
% Actual values are stored in the file: pvalues.mat

%  p0 = [1.0, 1.0, 1.0, 1.0];  % Before Switch
%  p1 = [1.0, 0.4, 1.0, 0.2];  % After Switch
%  y0 = [1.0, 0.1];            % Initial Condition
%  ft = 3;                     % Final time
%  st = 2;                     % Switch time
%  errv = 0.05;                % Measurement Error
%  dt = 0.10;                  % Discretization Step

load pvalues;                  % load stored parameters
rand('seed',seed);             % initialize random generator

% Parameter Value Input

flag = 1;
while flag == 1,
     save pvalues;
     fprintf('\n\n                                    ');
     fprintf('SYSTEM EQUATIONS:\n\n');
     fprintf('  Time');
     fprintf('      x(0) = %4.2f', y0(1));
     fprintf('                y(0) = %4.2f\n\n', y0(2));
     fprintf(' [0, %g]    ', min(st,ft));
     fprintf(' dx/dt = %4.2fx - %4.2fxy    ', p0(1), p0(2));
     fprintf(' dy/dt = -%4.2f(y-1) + %4.2fxy\n\n', p0(3), p0(4));
     if st < ft,
        fprintf(' (%g, %g]    ', st,ft);
        fprintf(' dx/dt = %4.2fx - %4.2fxy    ',p1(1),p1(2));
        fprintf(...
               ' dy/dt = -%4.2f(y-1) + %4.2fxy\n\n',p1(3),p1(4));
     end;
```

```
      fprintf(' Measurement:');
      fprintf('  Sampling = %g', dt);
      fprintf('  Variance = %g', errv);
      fprintf('  Generator = %g\n\n',rand('seed'));
      flag = 0;
      q = input('Change System Coefficients [y/n] ? ', 's');
      if q == 'y',
         flag = 1;
         p0 = input('Before Change [p(1) p(2) p(3) p(4)] ? ');
         p1 = input('After Change  [p(1) p(2) p(3) p(4)] ? ');
      end;
      q = input('Change Time Points [y/n] ? ', 's');
      if q == 'y',
         flag = 1;
         st = input('Change Time ? ');
         ft = input('Final Time ? ');
      end;
      q = input('Change Initial Conditions [y/n] ? ', 's');
      if q == 'y',
         flag = 1;
         y0 = input('[x(0) y(0)] ? ');
      end;
      q = input('Change Measurement Parameters [y/n] ? ', 's');
      if q == 'y',
         flag = 1;
         errv = input('Variance ? ');
         dt = input('Sampling Interval ? ');
         newseed=input('Random Generator Seed = ? ');
         if newseed >= 0
            seed = newseed;
            rand('seed',seed);
         end;
      end;
end;

p__ = p0;        % p__ is a global variable to pass parameters
global p__;      % to volterra.m subroutine

flag = 1;
%
% Program "run levels" controlled by 'flag' variable:
% 0 = just run
% 1 = raw data pause
% 2 = raw data & filter pause
% 3 = add to 2 continuous data
% 4 = empty
% 5 = plot and print all, pause each time
%
fprintf('\n\n                                Please Wait ...\n');

% Schedule Model Simulation

model(y0,0.0,st,ft,p0,p1,dt,0.0,errv,flag);

                *** End of startup command file ***
```

```
   ************************************************************
   *  Dynamical System Simulation and Subroutines Scheduling  *
   ************************************************************

function model (y0,t0,t1,tf,teta0,teta1,delta,var0,var1,flag);
%
% model(y0, t0, t1, tf, teta0, teta1, delta, var0, var1, flag)
%
% y0 = initial conditions y0=[y0(1), y0(2)]=["cancer","immune"]
% t0 = initial time, t1 = change time, tf = terminal time
% teta0 = initial parameters, teta1 = parameters after the switch
% delta = discretization step
% var0 = initial estimation error
% var1 = measurement error
% flag = run levels (see start.m)
% this routine calls function 'volterra' to calculate dy/dt
%
   hold off;
%
%  before change
%
   p__ = teta0;
   [t,y] = ode23('volterra',t0,t1,y0);
   nt = length(t);
%
% after change
%
   if t1 < tf,
      p__ = teta1;
      y1 = y(nt,1:2);
      [ts,ys] = ode23('volterra',t1,tf,y1);
      ns = length(ts);
      t(nt+1:nt+ns) = ts;
      y(nt+1:nt+ns,1:2) = ys;
      nt = nt + ns;
   end

   ymin01 = min(y(1:nt,1));
   ymax01 = max(y(1:nt,1));
   ymin02 = min(y(1:nt,2));
   ymax02 = max(y(1:nt,2));
   ymin0 = min(ymin01,ymin02) - 3*var1;
   ymax0 = max(ymax01,ymax02) + 3*var1;

   if  flag > 4,  % plots phase plane
       plot(ymin01,ymin02,'i',ymax01,ymax02,'i');
       hold on;
       plot(y(1:nt,1),y(1:nt,2),'-');
       hold off;
       pause;
   end

   if flag > 0,
      plot(t0,ymin0,'i',tf,ymax0,'i');
      grid
      hold on
   end
```

```
   if flag > 2,
      plot(t,y(1:nt,1),'-');
      title('--- = Cancer Cells    - - = Immune Response');
      plot(t,y(1:nt,2),'--');
      pause;
   end
%
%  Call Conditionally Linear Filter Routine:
%
   [miu,tmiu]=...
   clfilter(t0,t1,tf,teta0,delta,var0,var1,t,y,nt,flag);

   if  flag > 0
       title(sprintf(...
       'Measurement => Variance = %g  Sampling = %g',...
                      var1, delta));
       pause;
   end
   hold off;
%
%  Call Change Detection Routine
%
   [ta,tc] = seqtest(t0,tf,tmiu,miu,flag);

   hold off;
   plot(t0,ymin0,'i',tf,ymax0,'i');
   grid
   hold on
   plot(t,y(1:nt,1),'-');
   plot(t,y(1:nt,2),'--');

   if ta > 0  &  tc > 0,
      title(sprintf('Detection = %g   Alarm = %g', tc, ta));
      ymeasured = sample(ta,t,y(1:nt,2));
      text(ta,ymeasured,setstr(97));
      ymeasured = sample(tc,t,y(1:nt,2));
      text(tc,ymeasured,setstr(100));
   else
      title('No Change is Detected');
   end

               *** End of 'model' subroutine ***


          ********************************************
          *  Conditionally Linear Filter Subroutine  *
          ********************************************

function [miu,tm]=...
         clfilter(t0,t1,tf,teta0,delta,var0,var1,t,y,nt,flag);
%
% Filter for Lotka-Volterra Model with corrective noise terms
%
% [miu,tm] =...
% clfilter(t0,t1,tf, teta0, delta, var0,var1, t,y, ts,ys, flag)
%
% calculated here are (normalized) innovations
```

```
% t0 = initial time, t1 = switch time, tf = terminal time
% teta0 = initial parameters
% delta = discretization step
% var0 = initial estimation error
% var1 = measurement error
% [t,y] = data, nt = data length
%
   rand('normal');

%  Filter Equations:
%
%  x(t) == y1(t) => xf(t)
%  y(t) == y2(t) => yf(t)
%
%  x(t+1) = F(y(t))*x(t) + G(x(t),y(t))*w(t+1)
%  y(t+1) = h(y(t))+H(y(t))*x(t) + R(x(t),y(t))*w(t+1) + Q*v(t+1)
%
%  F(y(t)) = exp(delta*(teta0(1)-teta0(2)*y(t)))
%  G(x(t),y(t)) =  delta*teta0(2)*x(t)*var1*F(y(t))
%  H(y(t)) = (1-exp(-delta*teta0(3)))*teta0(4)*y(t)/teta0(3)
%  h(y(t)) = (y(t)-1)*exp(-delta*teta0(3)) + 1
%  R(x(t),y(t)) = -exp(-delta*teta0(3))*var1 - ...
%              (1-exp(-delta*teta0(3))*teta0(4)*x(t)*var1/teta0(3)
%  Q = var1
%
%  ypred = h + H*xf(t)
%  B = 1/sqrt(Q*Q + R*R + H*H*Pf(t))
%  miu(t+1) = B * (y(t+1) - ypred)
%  K = (G*R + F*H*Pf(t))*B
%  xf(t+1) = F*xf(t) + K*miu(t+1)
%  Pf(t+1) = F*F*Pf(t) + G*G - K*K

if delta > 0

   xf = y(1,1) + var0*rand;
   Pf = var0*var0;
   yf = y(1,2) + var1*rand;
   Qcap = var1*var1;
   nsample = 0;

   c1 = delta * teta0(1);
   c2 = delta * teta0(2);
   c3 = exp(-delta * teta0(3));
   if teta0(3) == 0,
      c4 = delta * teta0(4);
   else
      c4 = (1-c3)*teta0(4)/teta0(3);
   end

   for ta = t0+delta: delta: tf,

       ymeasured = sample(ta,t,y(1:nt,2));
       nsample = nsample + 1;
       ysample = ymeasured + var1*rand;

       if flag > 0,
          plot(ta,ysample,'o');
       end
```

```
        Fcap = exp(c1 - c2 * yf);
        Gcap = Fcap * c2 * xf * var1;
        Hcap = c4 * yf;
        hlow = (yf - 1) * c3 + 1;
        Rcap = -(c3 + c4 * xf) * var1;

        ypred = hlow + Hcap*xf;
        Bcap = 1/sqrt(Qcap + Rcap*Rcap + Hcap*Hcap*Pf);

        miu(nsample) = Bcap * (ysample - ypred);
        tm(nsample) = ta;

        Kcap = (Gcap*Rcap + Fcap*Hcap*Pf)*Bcap;
        xf = Fcap*xf + Kcap*miu(nsample);
        Pf = Fcap*Fcap*Pf + Gcap*Gcap - Kcap*Kcap;
        if flag > 1
           plot(ta,ypred,'+');
           plot(ta,xf,'x');
        end
        yf = ysample;
    end
end

        *** End of Conditionally Linear Filter Routine ***


                  ***********************************
                  *  Change Detection Subroutine  *
                  ***********************************

function [ta,tc] = seqtest (t0, tf, t, y, flag);
% Sequential test for change from zero to a nonzero
% mean in a Gaussian sequence [t,y] with a known
% constant variance
%
% [ta, tc] = seqtest(t0, tf, t, y, flag)
%
% t0 = initial time, tf = terminal time
% [t,y] = data to be tested
%
% ta = alarm time
% tc = change detection time
%
   nt = length(t);

% load pre-calculated variables
  load therdis;
  if nt > nmax,      % not enough thresholds were pre-calculated
     fprintf(setstr(7));
     fprintf('nmax = %g  <  nt = %g  ?\n',nmax, nt);
     pause;
  end

  den(1) = 1;
  for k = 2 : nt,
      den(k) = 1 / sqrt(k);
  end
```

```
% sigma is a scalar multiplier which monitors false alarm rates
% higher value of sigma => smaller detection sensitivity
   sigma = 1.5;            % as high as 4.5 for sparse sampling
   thermax = therave + sigma*therdev;

   satur = thermax(nt);

   plot(t0,0,'i',tf,satur,'i');
   hold on

   naver = 3;       % length of alarm function averaging window
   eaver = 1;       % auxiliary monitor of change time estimation
                    % consistency
   for td = 1 : nt,
       maxtemp = -1;
       delta = 0;
       for tc = td : -1 : 1,
           delta = delta + min(satur,max(-satur,y(tc)));
           ratio = abs(delta)*den(td-tc+1);
           if ratio > maxtemp,
              ttemp = tc;
              maxtemp = ratio;
           end
       end

       if maxtemp > thermax(td),
          maxtemp = thermax(td);
          plot(t(td),maxtemp,'*');
       else
          plot(t(td),maxtemp,'+');
       end

       maxratio(td) = maxtemp;
       te(td) = t(ttemp);

       if td >= naver,
          saver = std(te(td-naver+1:td));
          raver = mean(maxratio(td-naver+1:td));
          if ((saver < eaver) & ...
              (raver >= 0.9*mean(thermax(td-naver+1:td)))),
              fprintf(setstr(7));
              ta = t(td);
              tc = te(td);
              hold off;
              return;
          end
       end
   end

   hold off

   ta = -1;
   tc = -1;
   return;

           *** End of Sequential Detection Routine ***
```

```
                    ****************************
                    *  Lotka-Volterra System   *
                    ****************************

function yprime = volterra(t,y);
% volterra(t,y) returns the derivatives of the Lotka-Volterra
% system of equations
% y'(1) = y(1) * (p(1) - p(2)*y(2))
% y'(2) = y(2) * (p(4)*y(1) - p(3)) + p(3) ¦ p(5)=1
% needs global parameters:
% p__ = [p__(1),p__(2),p__(3),p__(4)],
% normally set by the calling subroutine model.m

yprime = [ (y(1) .* (p__(1) - p__(2) .* y(2)));
           (y(2) .* (p__(4) .* y(1) - p__(3)) + p__(3)) ];

                *** End of Volterra Routine ***


             ****************************************
             *  Pre-Calculation of Alarm Thresholds  *
             ****************************************

% Creates and stores in c:\matlab\therdis.dat
% theoretical distribution parameters used by sequential
% change test routine
%
  clear
  clc
  nmax = input('Number of Thresholds = ? ');
  therave(1) = 1;
  for k = 2 : nmax,
      therave(k) = therave(k-1) + 1 / sqrt(k);
  end

  therdev(1) = 0;
  therdev(2) = 1;
  for k = 2 : nmax-1,
      S = 0;
      for i = 1 : k-1,
          S = S + 1 / sqrt(i*(k+1-i));
      end
      therdev(k+1) = therdev(k) + 1 / sqrt(k) + S;
  end

  for k = 1 : nmax,
      therdev(k) = sqrt(1+(therdev(k)-therave(k)^2)/(k*pi));
      therave(k) = therave(k) / sqrt(pi*k);
  end

  save c:\matlab\therdis therave therdev nmax;

       *** End of threshold pre-calculation routine ***

                ********************************
                *  End of Subroutine Listings  *
                ********************************
```

References

[1] W.J. Kolodziej, R.R. Mohler, "On structurally decomposing processes", *IEEE Trans. Circuits and Systems*, vol. 35, pp. 1417-1424, November, 1988.

[2] A.S. Willsky, "A survey of design methods for failure detection in dynamic systems", *Automatica*, vol. 12, pp. 601-611, 1976.

[3] A. Benveniste, M. Basseville, G.V. Moustakides, "The asymptotic local approach to change detection and model validation", *IEEE Trans. Aut. Cont.*, Vol. AC-23, No. 7, July 1987.

[4] M. Basseville, A. Benveniste, eds., "Detection of abrupt changes in signals and dynamical systems", *Lecture notes in Control and Information Sciences*, vol. 77, New York, Springer-Verlag, 1986.

[5] L. Le Cam, "Asymptotic Methods in Statistical Decision Theory", *Springer Series in Statistics*, New York, Springer-Verlag, 1986.

[6] G.I. Marchuk, "Mathematical Modelling in Immunology and Medicine", North-Holland, Amsterdam, 1983.

[7] A.L. Asachenkov, R.R. Mohler, "A Systems Overview of Immunology, Disease and Related Data Processing", IIASA Working Paper, WP-89-31, February 1989.

[8] R.S. Lipster, A.N. Shiryaev, "Statistics of random processes", New York, Springer Verlag, 1978.

[9] I.A. Ibragimov, R.Z. Has'minskii, "Statistical Estimation - Asymptotic Theory", *Aplications of Mathematics*, vol. 16, New York, Springer Verlag, 1981.

[10] P. Billingsley, "Convergence of Probability Measures", *Wiley Series in Probability and Mathematical Statistics*, New York, 1968.

[11] A.N. Shiryaev, "Optimal Stopping Rules", *Applications of Mathematics*, vol. 8, New York, Springer Verlag, 1978.

[12] W. Feller, "An Introduction to Probability Theory and Its Applications", vol. II. Wiley, New York, 1971.

[13] W.J. Kolodziej, R.R. Mohler, "State Estimation and Control of Conditionally Linear Systems", *SIAM J. Contr. Optimiz.*, 24, 1986.

[14] W.J. Kolodziej, A. Pacut, "Filtering and detection problems for nonlinear time series", in *Nonlinear Time Series and Signal Processing* (R. Mohler, ed.), New York, Springer Verlag, 1988.

[15] A.R. Gallant, H. White, "Estimation and Inference for Nonlinear Dynamic Models", Blackwell, New York, 1988.

[16] S. Karlin, H.M. Taylor, "A Second Course in Stochastic Processes", Academic Press, New York, 1981.

[17] L. Takacs, "Combinatorial Methods in the Theory of Stochastic Processes", Krieger Publishing, New York, 1967.

[18] F. Spitzer, "Principles of Random Walk", Van Nostrand-Reinhold, New York, 1964.

[19] F.R. Hampel et al., "Robust Statistics - The Approach Based on Influence Functions", Wiley Series in Probability and Mathematical Statistics, Wiley, 1986.

[20] P.J. Huber, "Robust Statistics", *Wiley Series in Probability and Mathematical Statistics*, 1981.

[21] P.J. Huber, "Robust Statistical Procedures", *Regional Conference Series in Applied Mathematics*, Society for Industrial and Applied Mathematics, 1977.

[22] A. Wald, "Sequential Analysis", Dover Publications, New York, 1947.

[23] W.J. Kolodziej, A. Pacut, "Robust Detection of Structural Changes in Nonlinear Stochastic Systems", Proceedings of IEEE ISCAS, Portland, Oregon, May 1989.