Fast fault diagnosis for industrial processes applied to the reliable operation of robotic systems

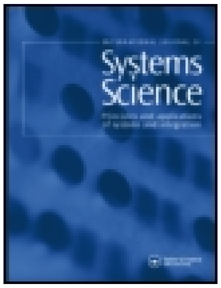
Article *in* International Journal of Systems Science · July 1989

DOI: 10.1080/00207728908910209

2 authors:

Anastasios D. Pouliezos
Technical University of Crete

G. Stavrakakis
Technical University of Crete



International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

Fast fault diagnosis for industrial processes applied to the reliable operation of robotic systems

A. POULIEZOS [a] & G. S. STAVRAKAKIS [a]
[a] Technical University of Crete, GR-731 00, Chania, Greece
Published online: 01 Feb 2007.

INT. J. SYSTEMS SCI., 1989, VOL. 20, NO. 7, 1233–1257

Fast fault diagnosis for industrial processes applied to the reliable operation of robotic systems

A. POULIEZOS† and G. S. STAVRAKAKIS†

Fault detection, identification and monitoring play a primary role in systems engineering. This also holds for advanced processes, such as robotic systems, with highest demands on reliability and safety. A method is presented for fast fault detection and location, due to the data deweighting in the identification procedure of the monitored process parameters and the iterative on-line calculation of the statistics of the fault detection scheme. This method has many advantages for microcomputer applications and guarantees a very early process fault detection. Application of the method to the fast fault detection of the dc motor actuators of a robotic system is described.

Notation

N_i, K_{mi}, R_i, L_i ($i = 1, 2, 3$)	are the gear ratio, the dc torque constant, the armature resistance, the armature inductance of the dc-motors, respectively
V_i	is the armature voltage (control input) of the ith dc-motor
ρ_i	is the viscous friction coefficient of the ith dc-motor
J_{mi}	is the moment of inertia of the rotor of the ith dc-motor
$\dot{\omega}_{mi} = N_i \dot{q}_i = N_i \omega_i$	is the reflected shaft angular velocity
$A(q)$	is the 3×3 generalized inertial matrix which is symmetric and positive-definite
$\Gamma(q)$	is the 3×1 vector of the gravitational torques
$D(q)h(\dot{q})$	involves the centrifugal and the Coriolis torques (with $h(\dot{q})$ = vector of $\dot{q}_i \dot{q}_j$, $i = 1, 2, 3$, $j = i, \ldots, 3$)
$i_i(t)$	is the armature current of the ith joint actuator
$T_i(t) = K_{mi} N_i i(t)$	is the reflected electromagnetic torque of the ith joint actuator
$T_{Li}(t) = T_i(t) - J_{mi} N_i \omega_{mi} - \rho_i N_i \omega_{mi}$	is the torque generated at the ith joint (reflected load of the ith dc motor)

Received 8 February 1988.

† Technical University of Crete, GR-731 00 Chania, Greece.

0020-7721/89 $3.00

1. Introduction

Increasing demands on reliability and safety of industrial processes and their elements led to the development of methods for improving supervision and monitoring as part of the overall control of the process. This is also true for fine processes with highest demands on reliability and safety, e.g. robotic manipulators (see Nicosia and Tomei 1984, Tzafestas and Stavrakakis 1986, Paul 1981, Pouliezos *et al.* 1986).

In the next sections, the elementary functions of process supervision are considered, as they are given by Isermann (1984) and Geiger (1986). Figure 1 shows a block diagram for process supervision based on parameter estimation.

The basis of this class of methods is the combination of theoretical modelling and

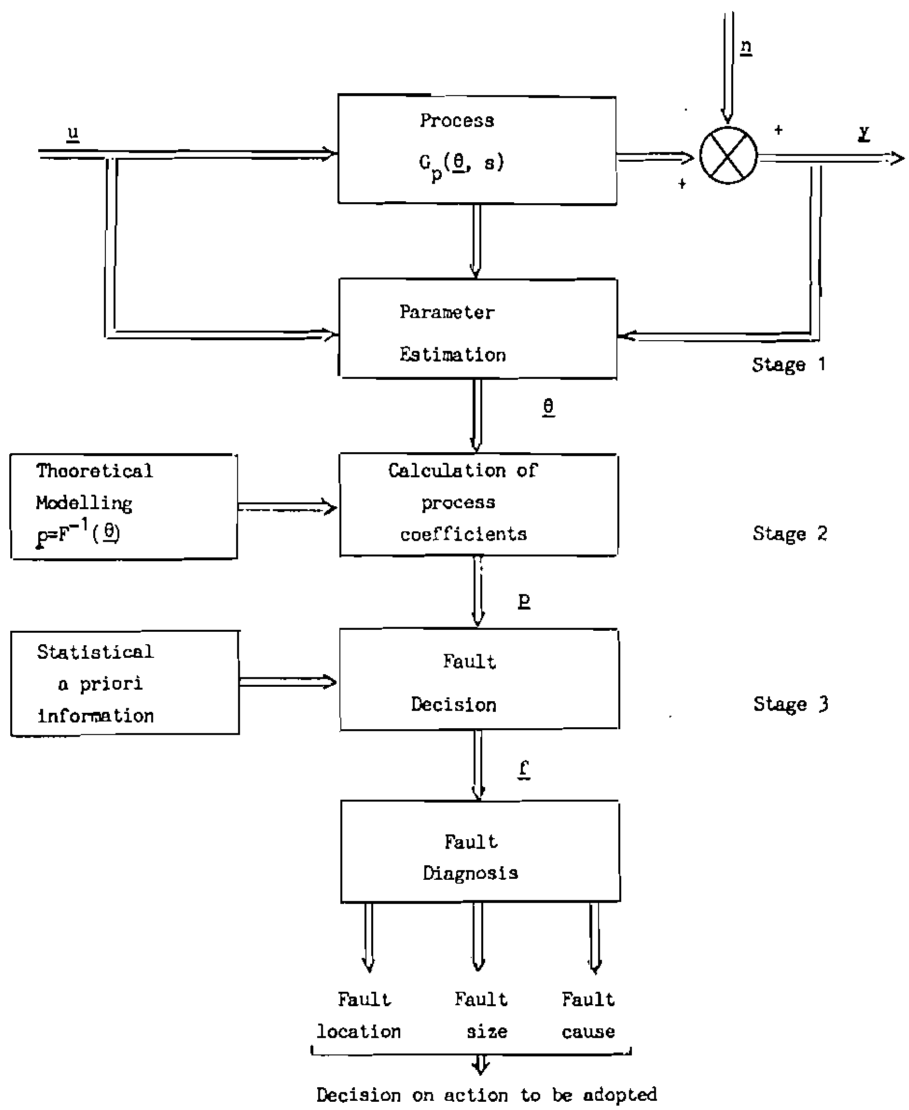


Figure 1. Generalized structure of fault detection method based on parameter estimation and theoretical modelling.

parameter estimation of continuous-time models. A necessary requirement of this procedure is, however, the existence of the inverse relationship $p = F^{-1}(\theta)$ when the relationship between the model parameters θ_1 and the physical process coefficients p_1, $\theta = f(p)$ is well determined. Therefore, it may be restricted to well-defined processes (see Isermann 1984). In the case of the robotic systems studied in this paper, the dc motor or hydraulic actuators are well-defined processes and theoretical models can be found in almost all the robot systems control literature (Nicosia and Tomei 1984, Tzafestas and Stavrakakis 1986, Paul 1981, Pouliezos *et al.* 1986).

A fault is to be understood here as a non-permitted deviation of a physical parameter from the nominal value, which leads to the inability to fulfill the intended purpose. After the effect of a fault is known, a decision on the action to be taken can be made. If the evaluated fault is tolerable, the operation may continue and if it is conditionally tolerable a change of operation has to be applied. However, if the fault is not tolerable, the operation must be stopped immediately and the fault eliminated. If a process fault appears, it has to be detected as early as possible.

When the physical process coefficients p, which indicate process faults, are not directly measurable, an attempt can be made to determine their changes via the changes in the process model parameters θ (see Fig. 1 and Isermann 1984). The estimation of the model parameters θ_i can result from the measurements of the signals $y(t)$ and $u(t)$, the process equation (derived by theoretical modelling) for the measurable input and output variables $y(t) = f(u(t), \theta)$ and modern estimation theory.

A common occurrence is that data are received sequentially over time. Therefore, improved recursive discrete-time estimation techniques should be applied for better parameter estimation performance.

After parameter estimation (stage 1) and calculation of the physical process coefficients $p(k)$ (stage 2), follows the fault identification procedure (stage 3). By fault detection is meant the algorithm for estimating no-fault and faulty case statistics of the monitored parameters in order to detect the exact instant of the fault occurrence and the size of the fault. In this paper a method to calculate iteratively the exact sample correlation for sequentially received data is given and proved.

These improved estimation and fault detection methods permit very early process fault detection for general on-line microcomputer applications. The methods and applications for fault detection and location for the dc motor actuators of robot systems are described and simulation results are given.

2. Dynamic models for the MRAC robotic manipulator

Many results on the control of robotic manipulators have been derived and widely disseminated. Among the control algorithms developed, the so-called model reference adaptive control (MRAC) approach seems to provide a robust method for the control of processes with variable parameters, and/or unknown parts. The MRAC model assures convergence to suitable reference models for a class of processes, one of which is the manipulator.

Nicosia and Tomei (1984) use the complete non-linear time-varying model including all second-order terms, resulting from a generalized application of langrangian dynamics, to derive a robust MRAC algorithm for the manipulator based on Popov's hyperstability theory. In their paper, the dc-motor actuator dynamics are neglected resulting in sudden fluctuations for the derived control torques.

Tzafestas and Stavrakakis (1986) extended this original approach by introducing the dc-motor dynamics with some improvement concerning the on-line calculation of the controlled parameters. The results can be compared with those obtained by Nicosia and Tomei (1984). It is easy to see that if the dc-motor dynamics are taken into account for the control voltage calculation, the control voltage is smooth and easier to apply in practice than the control torque proposed by Nicosia and Tomei (1984). The global dynamic model of a 3-DOF (degrees of freedom) robotic manipulator is derived based on the lagrangian approach and the dynamic equations for the armature circuit and the mechanics of the dc-motor actuators. These equations have the following form (see Fig. 2)

$$\begin{bmatrix} \dot{T} \\ \dot{\omega} \end{bmatrix} = -G^*(q)\{C^*(q)h^*(\omega, T) - \Gamma^*(q)\} + G^*(q)SV \tag{1}$$

where

$$q^{\mathrm{T}}(t) = [q_1(t) \quad q_2(t) \quad q_3(t)]$$

$$V^{\mathrm{T}}(t) = [V_1(t) \quad V_2(t) \quad V_3(t)]$$

$$\omega^{\mathrm{T}} = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]$$

$$G^*(q) = \begin{bmatrix} K_{mN} & 0 \\ 0 & G(q) \end{bmatrix} \in R^{6\times 6}$$

$$C^*(q) = \begin{bmatrix} 0 & I & (K_{mN})^{-1}K_{RL} \\ D(q) & RN^2 & -I \end{bmatrix} \in R^{6\times 12}$$

$$h^*(\omega, T) = \begin{bmatrix} h(\omega) \\ \omega \\ T \end{bmatrix} \in R^{12}, \quad \Gamma^*(q) = \begin{bmatrix} 0 \\ \Gamma(q) \end{bmatrix} \in R^6$$

$$S = \begin{bmatrix} -(K_{mN})^{-1}K_{mL} \\ 0 \end{bmatrix} \in R^{6\times 3}$$

with

$$K_{mN} = \operatorname{diag}\left\{\frac{(K_{mi})^2(N_i)^2}{L_i}\right\}$$

$$K_{RL} = \operatorname{diag}\left[\frac{R_i}{L_i}\right], \quad N = \operatorname{diag}[N_i]$$

$$R = \operatorname{diag}[\rho_i]$$

$$K_{mL} = \operatorname{diag}\left[\frac{K_{mi}N_i}{L_i}\right]$$

$$J_m = \operatorname{diag}[J_{mi}], \quad i = 1, 2, 3$$

$$G(q) = [A(q) + J_m N^2]^{-1}$$

The MRAC algorithm which provides a robust trajectory for robots using dc-motor actuator dynamics is fully presented by Tzafestas and Stavrakakis (1986).

The value of the dc-motor constants play an important role for the calculation of the control law at any instant. On the other hand, the correct operation of the whole robot depends strongly on the correct operation of the dc-motor actuators, i.e. on the conservation of their physical properties (parameters) during operation. Thus, sudden changes, modelled as faults, in the process parameters may lead to degradation of performance and even to instability. Sudden changes of the physical parameters of the actuators must, therefore, be automatically monitored in order to achieve a reliable and safe operation of the manipulator.

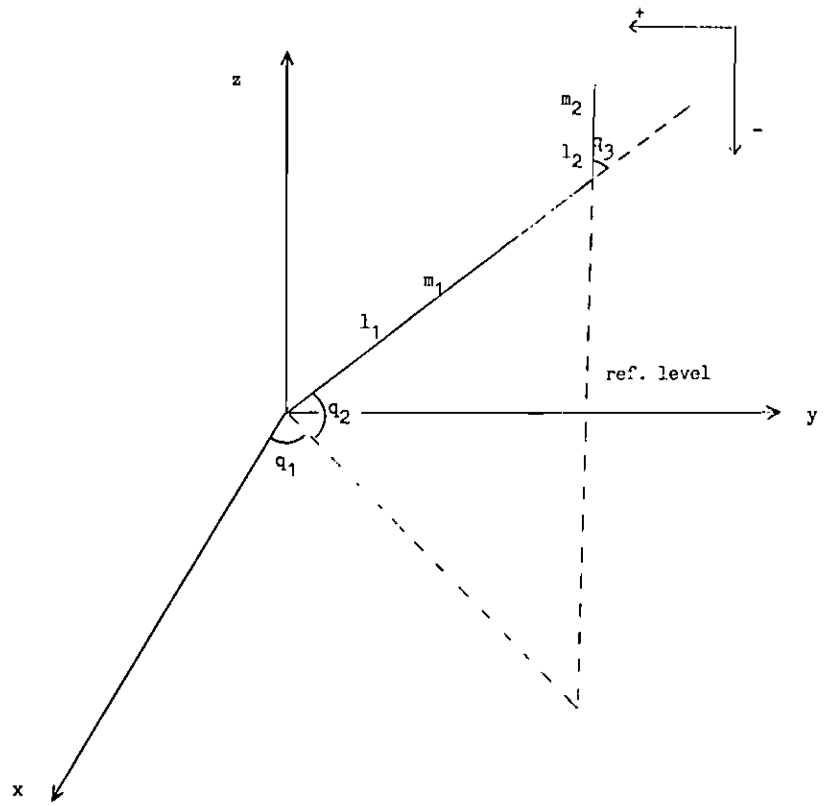


Figure 2. Geometry of the three-link manipulator under study.

3. Fast fault monitoring for the dc-motor actuators of a robot

The MRAC robotic system described in the previous section must be supervised automatically. The first stage of this supervision (see Fig. 1) consists of the detection of changes (faults) in the dc motor actuators based on theoretically derived motor models and parameter estimation. After the effect of the fault is known, a decision on the action to be taken can be made.

In many cases, the following variables may be assumed to be given:

(*a*) measurements of the input $y(t)$ and the output $u(t)$ of the process;

(*b*) more or less exact *a priori* information about the static and dynamic behaviour of the process.

3.1. *Mathematical model for the dc-motor actuator*

The dynamic model for the dc-motor actuator is given analytically in Appendix A. Define

$$u^{\mathrm{T}}(t) = [V_i(t) \quad T_{Li}(t)], \quad y(t) = [i_i(t) \quad \omega_i(t)]^{\mathrm{T}}$$

Then the model (A_1) of Appendix A can be formulated as a continuous-time multi-input multi-output (MIMO) system of two ($r = 2$) differential equations, for the actuator of each link; that is

$$y^{(1)}(t) + A_1 y^{(1)}(t) + A_0 y(t) = B_0 u(t) \tag{2}$$

where

$$A_1 = 0 \in R^{4\times 4}, \quad A_0 = \begin{bmatrix} \dfrac{R_i}{L_i} & \dfrac{K_{mi}N_i}{L_i} \\ -\dfrac{K_{mi}}{J_{mi}N_i} & \dfrac{\rho_i}{J_{mi}} \end{bmatrix}, \quad B_0 = \begin{bmatrix} \dfrac{1}{L_i} & 0 \\ 0 & -\dfrac{1}{J_{mi}(N_i)^2} \end{bmatrix}$$

Define

$$\theta_1 = \frac{R_i}{L_i}, \quad \theta_2 = \frac{K_{mi}N_i}{L_i}, \quad \theta_3 = \frac{1}{L_i}, \quad \theta_4 = -\frac{K_{mi}}{J_{mi}N_i}$$

$$\theta_5 = \frac{\rho_i}{J_{mi}}, \quad \theta_6 = -\frac{1}{J_{mi}(N_i)^2}$$

that is

$$\theta^{\mathrm{T}} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6] \in R^6$$

Then, if measurements of $y(t)$ and $u(t)$ are available, (2) leads to

$$e(t) = y^{(1)}(t) - \Psi(t)\theta$$

with

$$\Psi(t) = \begin{bmatrix} -y^{\mathrm{T}}(t) & u_1(t) & 0 & 0 \\ 0 & 0 & -y^{\mathrm{T}}(t) & u_2(t) \end{bmatrix} R^{2\times 6}$$

3.2. *Process parameter estimation using data deweighting*

In this stage, measurements of the input and output signals are considered to be available at discrete times $t = kT_0$, $k = 1, 2, \ldots, N$ with T_0 the sampling time. This means that measurements of $i_i(k)$, $\omega_i(k)$, $V_i(k)$, $T_{Li}(k)$, $k = 0, 1, \ldots, N$ are available.

In the present case of the 3-DOF MRAC robot these measurements are obtained through simulation (numerical integration) of the complete robot model (1), where the input voltage $V(t)$ is determined via the control algorithm proposed by Tzafestas and Stavrakakis (1986) in order to follow the desired trajectories.

The parameter estimation requires the first-order time derivatives of the noisy output signal $y(t)$. These derivatives can be calculated from the sampled measurements $y(k)$ using numerical differentiation. The simplest way is to replace the derivatives by the corresponding (backward) differences. To reduce the influence of the noise interpolation formulae, interpolation by third-order polynomials or Newton interpolation can be used. For calculating higher order derivatives, the use of state variable filtering is recommended (Isermann 1984, Geiger 1986). Suitable parameter estimation algorithms and their conditions for closed-loop applications are treated by Isermann (1982).

In almost all practical estimation problems, data are received sequentially over time. As each new group of data is received, it is used iteratively to improve the estimate of the unknown parameters p.

The outstanding numerical properties of the recursive discrete square-root filtering (DSF) and the classical RLS, provides a practical method for the detection of process parameter changes (Geiger 1986). However, if the iterative estimation process is carried out over a sufficient number of steps, one finds that the estimation error covariance matrix has decreased to very small values due to the repeated addition of a non-negative term to its inverse (Isermann 1982). This in turn causes the value of the estimator gain to become small.

This means that the corrections made to the parameter vector p_k in order to determine p_{k+1}, become small independent of what new or surprising information may be contained in the latest measurements. In order to prevent the recursive estimator from failing to respond adequately to new data, some form of data deweighting is often used.

The weighted least-squares method suggests that this can be realized by minimizing weighted estimation error squares. This leads to an exponential forgetting memory and the well-known RLS method with forgetting factor (see Isermann 1982).

These concepts can be extended to the DSF method described by Geiger (1986) through a lemma proved in Appendix B. These two methods are applied to the fast detection of the parameter changes of the dc motor robot actuators.

Let us define the error equation for MIMO-continuous time systems, formulated by a set of r-differential equations, as

$$e(t) = y^{(n)}(t) + \sum_{i=0}^{n} A_i y^{(i)}(t) - \sum_{w=0}^{m} B_w u^{(w)}(t) \tag{4}$$

with process input and output vectors $u(t) \in R^p$ and $y(t) \in R^r$, respectively, and parameter matrices $A_i \in R^{r \times r}$ and $B_w \in R^{r \times p}$, $p \leqslant r$. Observing the MIMO-system over k samples and storing the measurement vectors and matrices obtained from the continuous time system at discrete times, leads to

$$E(k) = Y^{(n)}(k) - \Phi(k)\theta \equiv D(k)\begin{bmatrix} -\theta \\ 1 \end{bmatrix} \tag{5}$$

with

$$D(k) = [\Phi(k) \quad Y^{(n)}(k)] \in R^{kr \times (rl+1)}$$

$$E^{\mathrm{T}}(k) = [e^{\mathrm{T}}(1) \quad \dots \quad e^{\mathrm{T}}(k)] \in R^{kr}$$

$$[Y^{(n)}(k)]^{\mathrm{T}} = [[y^{(n)}(1)]^{\mathrm{T}} \quad \dots \quad [y^{(n)}(k)]^{\mathrm{T}}] \in R^{kr}$$

$$\Phi(k) = \begin{bmatrix} \Psi(1) \\ \vdots \\ \Psi(k) \end{bmatrix} \in R^{kr \times rl}, \quad l \equiv (n+1)r + (m+1)p$$

$$\Psi(q) = \begin{bmatrix} [\Psi_1(q)]^T & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & \dots & [\Psi_i(q)]^T & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & \dots & [\Psi_r(q)]^T \end{bmatrix} \in R^{r \times rl}$$

where 0 is null vector $\in R^l$, $q = 1, 2, \dots, k$

$$\begin{aligned} [\Psi_i(q)]^T &= [-[y(q)]^T \quad -[y^{(1)}(q)]^T \quad \dots \quad -[y^{(n)}(q)]^T \\ &\qquad [u(q)]^T \quad [u^{(1)}(q)]^T \quad \dots \quad [u^{(m)}(q)]^T] \in R^l \\ q &= 1, 2, \dots, k \\ \theta^T &= [\theta_1^T \quad \theta_2^T \quad \dots \quad \theta_r^T] \in R^{rl} \\ \theta_j^T &= [(a_0^j)^T \quad \dots \quad (a_n^j)^T \quad (b_0^j)^T \quad \dots \quad (b_m^j)^T] \in R^l \\ j &= 1, 2, \dots, r \end{aligned}$$

and $(a_i^j)^T \in R^r$ is the jth row vector of the matrix A_i, $(b_w^j)^T \in R^p$ is the jth row vector of the matrix B_w and $i = 0, 1, \dots, n$, $j = 1, 2, \dots, r$, $w = 0, 1, \dots, m$.

The process parameters θ in (5), that is all the elements of the matrices of the MIMO-system (4), can be estimated if they are not constant but slowly time-varying by minimizing the weighted squares euclidean norm

$$V_e(k) = \|E_e(k)\|^2 \equiv \sum_{i=1}^{k} \varepsilon^2(i) e^T(i) e(i) \equiv D_e(k) \begin{bmatrix} -\theta \\ 1 \end{bmatrix} \tag{6}$$

with

$$\begin{aligned} [E_e(k)]^T &\equiv [\varepsilon(1) e^T(1) \quad \dots \quad \varepsilon(k-1) e^T(k-1) \quad \varepsilon(k) e^T(k)] \\ D_e(k) &\equiv [\Phi_e(k) \quad Y_e^{(n)}(k)] \\ \Phi_e(k) &= \begin{bmatrix} \varepsilon(1)\Psi(1) \\ \vdots \\ \varepsilon(k)\Psi(k) \end{bmatrix} \\ [Y_e^{(n)}(k)]^T &= [\varepsilon(1)[y^{(n)}(1)]^T \quad \dots \quad \varepsilon(k)[y^{(n)}(k)]^T] \end{aligned}$$

The choice of

$$\varepsilon(i) = \lambda^{(k-i)}, \quad 0 < \lambda < 1 \tag{6 a}$$

leads to an exponential forgetting memory. The forgetting factor λ is a constant, and it has to be chosen within $0{\cdot}95 \leqslant \lambda \leqslant 0{\cdot}995$ for most cases (see Isermann 1982).

Lemma 3.2

The data deweighting discrete square root filter (D-DSF) estimator of the slowly time-varying parameters θ of the MIMO-system (4) can be given in a recursive manner by the following algorithm:

(*a*) Start with $D_u(0) = 0$, which is the null $(rl + 1) \times (rl + 1)$ matrix.

For $k = 0, 1, 2, \ldots, N, \ldots$

(*b*) Find an orthogonal transformation matrix $T(k + 1)$ such that

$$T(k+1)\begin{bmatrix} \lambda D_u(k) \\ \hdashline z(k+1) \end{bmatrix} = \begin{array}{c} \overbrace{}^{rl+1} \\ \begin{bmatrix} D_u(k+1) \\ 0 \end{bmatrix} \end{array} \begin{array}{l} \}\, rl+1 \\ \}\, r \end{array} \tag{7}$$

where $D_u(k + 1) \in R^{(rl+1)\times(rl+1)}$ is an upper triangular matrix. The matrix $z(k + 1)$ is defined as

$$z(k+1) = [\Psi(k+1) \quad y^{(n)}(k+1)] \in R^{r\times(rl+1)} \tag{8}$$

and it is the new group of data. The matrix triangularization required in this step can be accomplished directly by one of the two most promising algorithms, Householder transformation and modified Gram–Schmidt (MGS) orthogonalization (see Kaminski *et al.* 1971).

(*c*) The upper diagonal $D_u(k + 1)$ matrix is partitioned as

$$D_u(k+1) = \begin{bmatrix} D_{\theta\theta}(k+1) & D_{\theta E}(k+1) \\ 0 & D_{EE}(k+1) \end{bmatrix} \tag{9}$$

where $D_{\theta\theta}(k + 1) \in R^{rl\times rl}$, $D_{\theta E}(k + 1) \in R^{rl\times 1}$, $D_{EE} \in R$ and 0 is an rl null vector.

The minimization of (6) leads to the following estimator of θ

$$\theta(k+1) = D_{\theta\theta}^{-1}(k+1)D_{\theta E}(k+1) \tag{10}$$

and the minimum value of $V(k + 1)$ is given by the value of $D_{EE}(k + 1)$, for $k = 1, 2, \ldots, N, \ldots$.

(*d*) Go back to step (*b*) and continue the procedure as new data is available.

Choosing the appropriate T may be interpreted as 'compressing' the deweighting data matrix $D_E(k)$ into the upper triangular matrix $D_u(k)$.

3.3. *Calculation of the process physical coefficients*

The physical process coefficients of the simulated robot dc-motor actuators were selected to be

$$\begin{array}{lll} p_1 = R_i & p_4 = J_{mi} & V_{max} = 12V \\ p_2 = L_i & p_5 = \rho_i & \\ p_3 = K_{mi} & N_i = 64 & \end{array}$$

The relationship between process model parameters and physical process coefficients p is given by

$$p_1 = \frac{\theta_1}{\theta_3} \tag{11 a}$$

$$p_2 = \frac{1}{\theta_3} \tag{11 b}$$

$$N_i p_3 = \frac{\theta_2}{\theta_3} \tag{11 c}$$

$$N_i^2 p_4 = -\frac{\theta_2}{\theta_3 \theta_4} \tag{11 d}$$

$$N_i^2 p_5 = -\frac{\theta_2 \theta_5}{\theta_3 \theta_4} \tag{11 f}$$

The case of a fault occurrence into the gear box is considered as an event with probability 0. Under this consideration the relations (11 a)–(11 f) hold for the physical process coefficients. However, the case of a fault in the gear box, i.e. modification of the no-fault value of the gear ratio can be detected via the fault occurrence or not in the parameters p_3, p_4, p_5 of the fourth stage.

The physical process coefficient vector p is used in stage 3 of the on-line fault identification method as input data.

4. Fault detection and identification for the robot dc-motor actuator

Fault detection and identification involves the decision that a fault has occurred (fault diagnosis), the localization of the fault and its cause and the estimation of the fault size.

After calculation of the physical process coefficients $p(k)$ in stage 2, let us consider $p(k)$ as a gaussian vector with its components statistically independent and its realizations $p(i)$ and $p(j)$ in the different sample instants $i <> j$ statistically independent. It is assumed that the mean vector $\mu(k) = E[p(k)]$ and the covariance matrix

$$C(k) = E[[p(k) - \mu(k)][p(k) - \mu(k)]^{\mathrm{T}}]$$

are invariant for the non-error case, i.e.

$$\mu(k) = [\bar{\mu}_1 \quad \dots \quad \bar{\mu}_m]^{\mathrm{T}}\text{: constant}$$

$$C(k) = \operatorname{diag}\{(\bar{\sigma}_1)^2, \dots, (\bar{\sigma}_m)^2\}\text{: constant}$$

Under these conditions the joint probability density function over N samples is defined as

$$f(p(1), \dots, p(N)) = \prod_{i=1}^{N} f(p(i)) \tag{12}$$

A fault is defined by a significant deviation of the mean μ_i and/or variance of $(\sigma_i)^2$ of $p_i(k)$, the ith component of $p(k)$, from the non-error value $\bar{\mu}_i$, $(\bar{\sigma}_i)^2$. Moreover, it is assumed that only one fault may occur at a time.

This is a classical hypothesis test problem and can be handled by the formulation of $(m + 1)$ hypotheses H_i, $0 < i < m$, where $m = 5$ in the case of the dc-motor of each robot joint actuator.

$$H_i = \begin{cases} \text{no fault in the mean and/or the variance of } \rho \ (i = 0) \\ \text{fault of type } i \text{ (significant deviation of mean and/or} \\ \text{variance of } p_i\text{), } 1 < i < m. \end{cases}$$

Each hypothesis H_i can be associated with a gaussian conditional density function, where $\mu(H_i)$ and $\sigma^2(H_i)$ denote conditional mean and variance for hypothesis H_i. Therefore, the non-error case is described by $\mu(H_0)$, $\sigma^2(H_0)$, whereas $\mu(H_i)$, $\sigma^2(H_i)$, $1 < i < m$, describe a fault of type i.

When the no-fault mean and variance are not known, i.e. the values for the parameters are not known *a priori*, an estimation algorithm of $\mu(H_0)$ and $\sigma^2(H_0)$ is needed.

The output of the fault identification stage is a fault vector

$$F = [i \quad \hat{\mu}_i(H_i) \quad \hat{\sigma}_i(H_i)]^{\mathrm{T}}, \quad 1 < i < m \tag{13}$$

where i represents the type of error, characterized by $\mu(H_i)$ and $\hat{\sigma}_i(H_i)$. The fault detection and localization is possible by computing the logarithmic likelihood ratios. The algorithm used here for estimating the non-error statistics and the fault detection and localization is described by Geiger (1986), where a common window technique is used in order to calculate the relevant statistics of the fault occurrence. However, the procedure employed is non-iterative and this greatly increases computational time. In the present method, the relevant statistics are calculated iteratively using the following formulae (see Pouliezos 1980).

(i) For the non-error case, H_0

$$(a)\ \ \bar{\mu}_i(k) = \frac{1}{k}[(k-1)\bar{\mu}_i(k-1) + \hat{p}_i(k)], \quad i = 1, \ldots, m, \quad k = 1, \ldots, N_s \tag{14}$$

The algorithm is initialized by $\bar{\mu}_i(1) = \hat{p}_i(1)$.

$$(b)\ \ \bar{\sigma}_i^2(k) = \frac{k-2}{k-1}\bar{\sigma}_i^2(k-1) + \frac{1}{k}(\hat{p}_i(k) - \bar{\mu}_i(k-1))^2 \quad i = 1, \ldots, m, \quad k = 2, \ldots, N_s$$

The algorithm is self-initialized by

$$\bar{\sigma}_i^2(2) = \tfrac{1}{2}(\hat{p}_i(1) - \hat{p}_i(2))^2$$

The sample size, N_s, is chosen so that an accurate estimate for the above statistics is obtained.

(ii) For the error-case, H_i, $i = 1, \ldots, m$, three quantities are needed.

(*a*) Window mean

$$\begin{aligned}
\hat{\mu}_i(k) &= \frac{1}{N_w}\sum_{j=k-N_w+1}^{k} \hat{p}_i(j) \\
&= \hat{\mu}_i(k-1) - \frac{1}{N_w}[\hat{p}_i(k-N_w) - \hat{p}_i(k)] \\
&= \hat{\mu}_i(k-1) - \frac{1}{N_w}\gamma_i(k), \quad i = 1, \ldots, m
\end{aligned} \tag{15}$$

where $\gamma_i(k) = \hat{p}_i(k-N_w) - \hat{p}_i(k)$, $i = 1, \ldots, m$.

(*b*) Window variance

$$\begin{aligned}
\hat{\sigma}_i^2(k) &= \frac{1}{N_w}\sum_{j=k-N_w+1}^{k} [\hat{p}_i(j) - \hat{\mu}_i(k)]^2 \\
&= \hat{\sigma}_i^2(k-1) + \frac{1}{N_w}
\end{aligned}$$

$$\times \left[2\gamma_i(k)\hat{\mu}_i(k-1) - \frac{1}{N_w}\gamma_i^2(k) - \hat{p}_i^2(k-N_w) + \hat{p}_i^2(k) \right], \quad i = 1, \dots, m \tag{16}$$

$$(c)\ \tilde{\sigma}_i^2(k) = \frac{1}{N_w} \sum_{j=k-N_w+1}^{k} [\hat{p}_i(j) - \bar{\mu}_i]^2$$

$$= \tilde{\sigma}_i^2(k-1) - \frac{1}{N_w}$$

$$\times [\hat{p}_i^2(k-N_w) - \hat{p}_i^2(k) - 2\bar{\mu}_i\gamma_i(k)], \quad i = 1, \dots, m \tag{17}$$

The proof of this last recursion is given in Appendix C.

These three iterative schemes need a starting window of N_w sample values $\hat{p}_i$ in order to be initialized. The window size, N_w, is chosen so that reasonable rates for missed alarms and false detections are achieved. The advantage of using a moving window of sample parameter values $\hat{p}_i(k)$, $i = 1, \dots, m$ is in the improved speed of detection. This is offset, however, by an increasing false detection rate.

A fault in the ith parameter is declared at time k, if the quantity

$$\Lambda_i(k) = \frac{N_w}{2}\left[\frac{\tilde{\sigma}_i(k)}{\bar{\sigma}_i(k)} - 2\ln\left(\frac{\hat{\sigma}_i(k)}{\bar{\sigma}_i(k)}\right) - 1\right]$$

exceeds a predetermined threshold in M consecutive time instants. The threshold value and M may be chosen by simulation runs. Alternatively the comparable quantity may be

$$\Lambda_i(k) = \frac{\tilde{\sigma}_i(k)}{\bar{\sigma}_i(k)} - 2\ln\left(\frac{\hat{\sigma}_i(k)}{\bar{\sigma}_i(k)}\right)$$

with the threshold value modified accordingly.

5. Simulation results

The physical process coefficients of the simulated dc-motor robotic actuator are

$$R = 1{\cdot}04\ \Omega \qquad J_m = 0{\cdot}00005\ \mathrm{kgm}^2$$
$$L = 0{\cdot}00089\ \mathrm{H} \qquad \rho = 0{\cdot}005\ \mathrm{kgm}^2/\mathrm{s}$$
$$K_m = 0{\cdot}0224\ \mathrm{V\ s/rad} \qquad N = 64$$

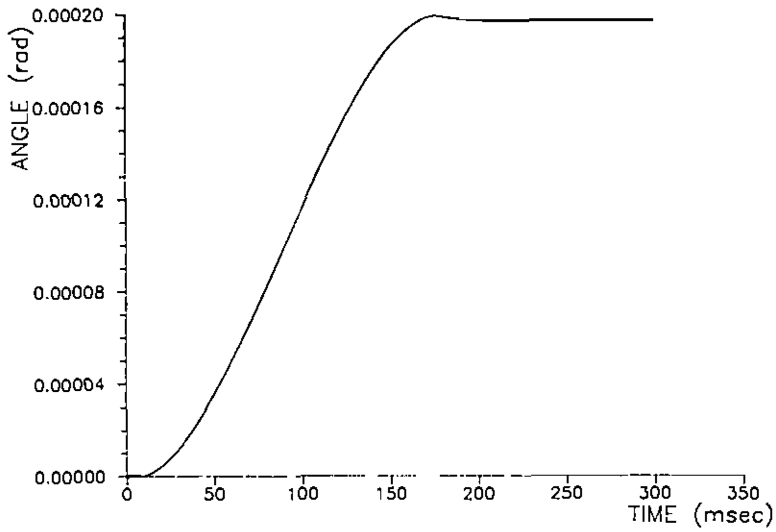
These values are in the range of the coefficients for dc-motor drives found in the literature. The proposed method was simulated on an IBM-PC microcomputer. The required derivatives for the model equations (see Appendix A) were calculated using a third-order backward formula

$$f'(x_0) = \frac{1}{2h}[3f_0 - 4f_{-1} + f_{-2}]$$

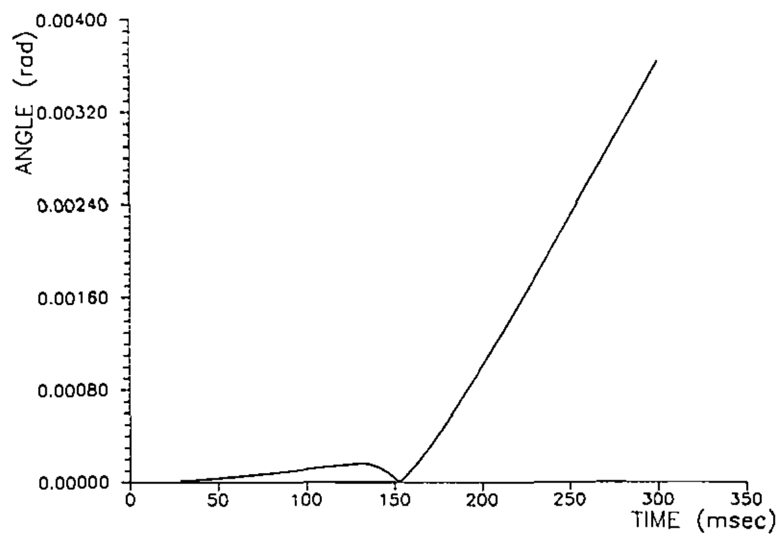
where h is the sampling interval. In the simulation, $h = 0{\cdot}0005$ s. The non-error statistics are calculated using $N_m = 300$ samples, whereas the detection window was $N_w = 50$. The first parameter estimate to be used by the detection procedure was taken at time $k = 70$, giving a large initial sample. The threshold value was 11·20 and $M = 10$.

Simulation tests were run for both estimation methods. The results are shown graphically in Fig. 3. In plot P1 the performance of the controller is seen in normal (no-fault) conditions. In plot P2 the same controller is graphed under fault conditions (a simulated rise in the motor resistance value from $1{\cdot}04\,\Omega$ to $1{\cdot}09\,\Omega$ at $t = 130$). Clearly a method for detecting such changes is necessary, since the system output (actual angle) is diverging away from the input (desired) angle.

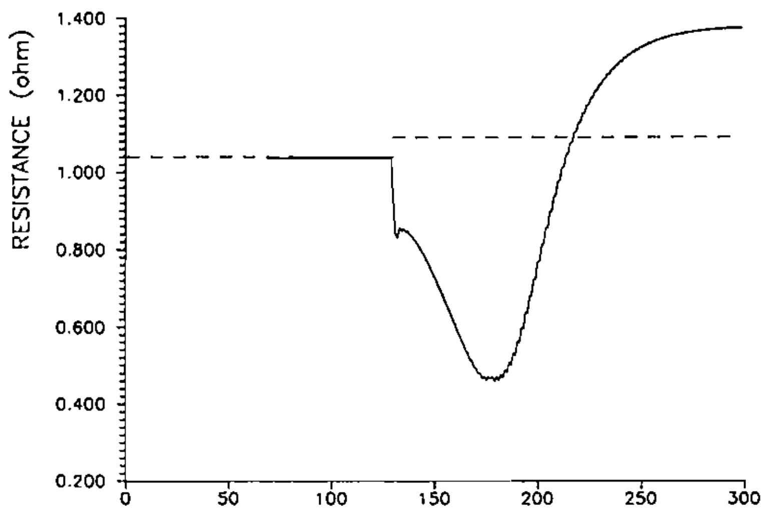
In plots P3(A)–(E), the output of the least-squares estimator using a forgetting factor matrix $B = \mathrm{diag}\,\{1, 1, 1, 1, 1, 1\}$ is shown. The actual parameter values are shown by the dotted line. The performance of the estimator is not satisfactory in this case, at least for the resistance and the inductance parameters. This may be explained by the fact that the estimator has infinite memory, so that data from the no-fault history influence the estimator.



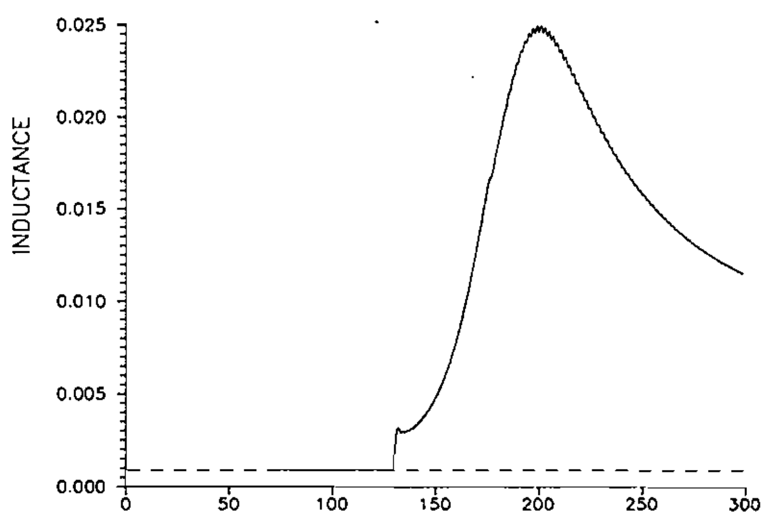
P1. Absolute difference of (desired-actual angle), no fault.



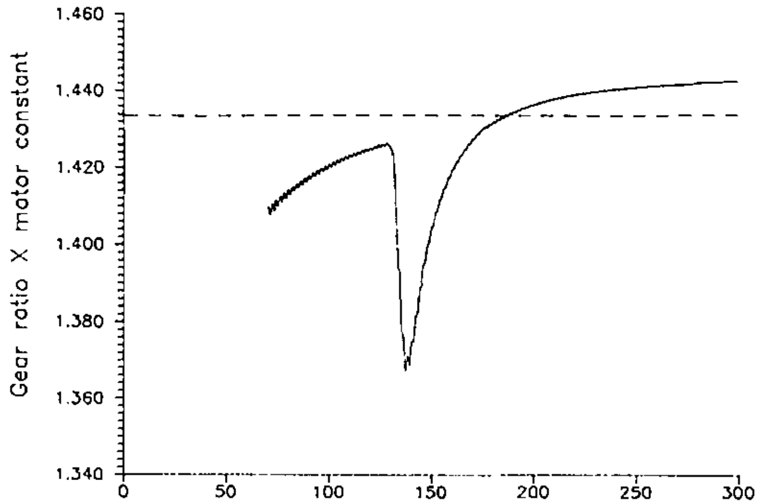
P2. Absolute difference of (desired-actual angle), fault at $t = 130$.


RESISTANCE (ohm)
1.400
1.200
1.000
0.800
0.600
0.400
0.200
0
50
100
150
200
250
300

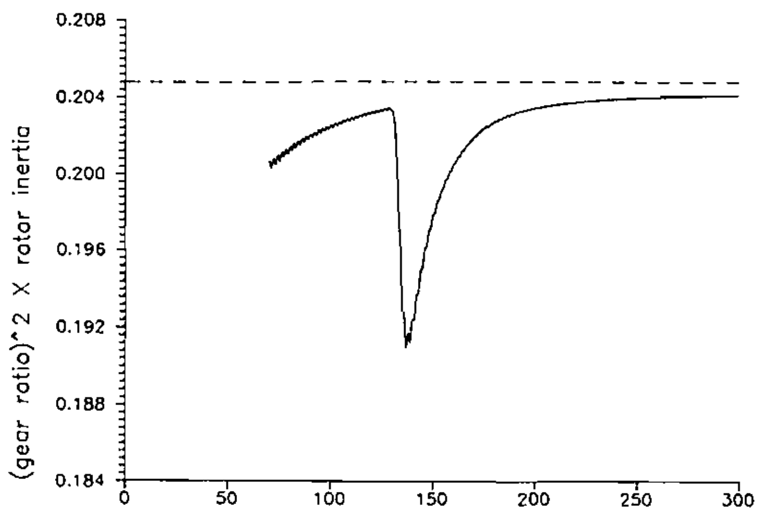
(A)


INDUCTANCE
0.025
0.020
0.015
0.010
0.005
0.000
0
50
100
150
200
250
300

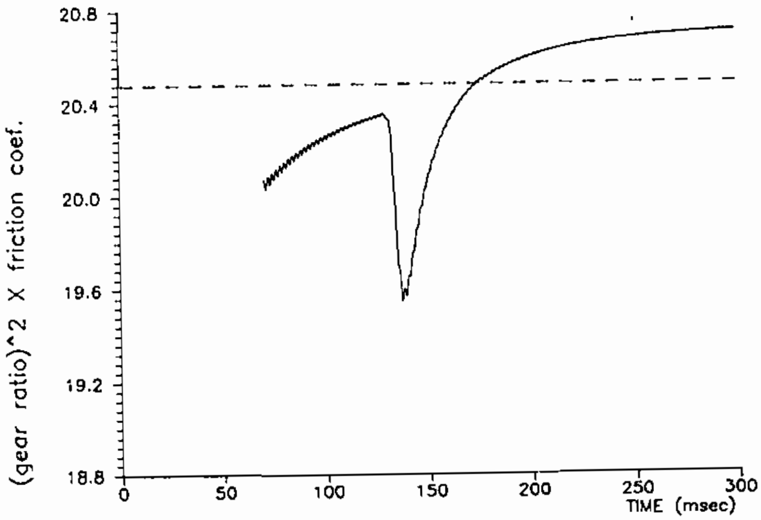
(B)


Gear ratio X motor constant
1.460
1.440
1.420
1.400
1.380
1.360
1.340
0
50
100
150
200
250
300

(*C*)

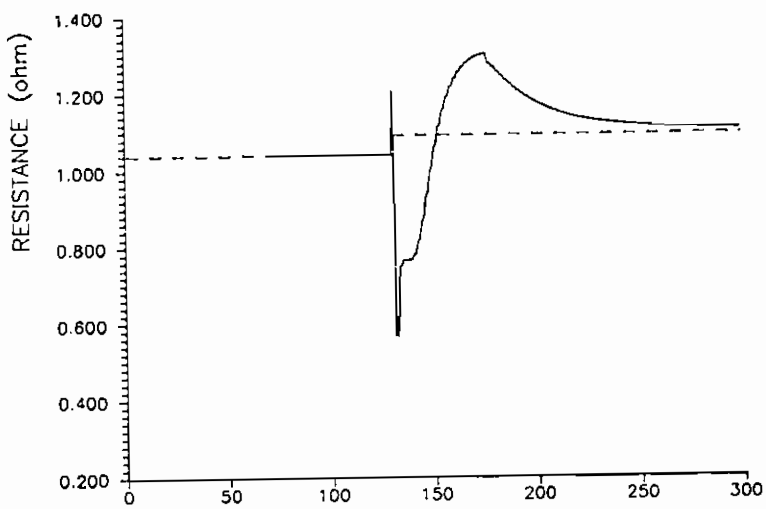

(gear ratio)^2 X rotor inertia
0.208
0.204
0.200
0.196
0.192
0.188
0.184
0
50
100
150
200
250
300

(*D*)

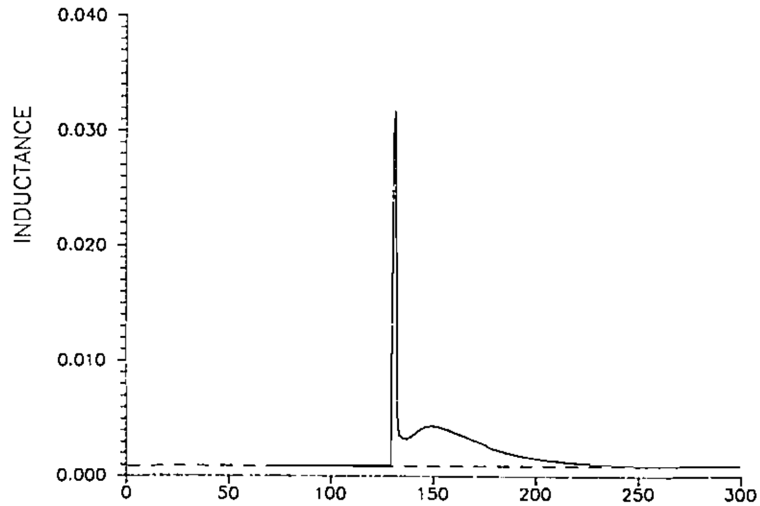


(E)

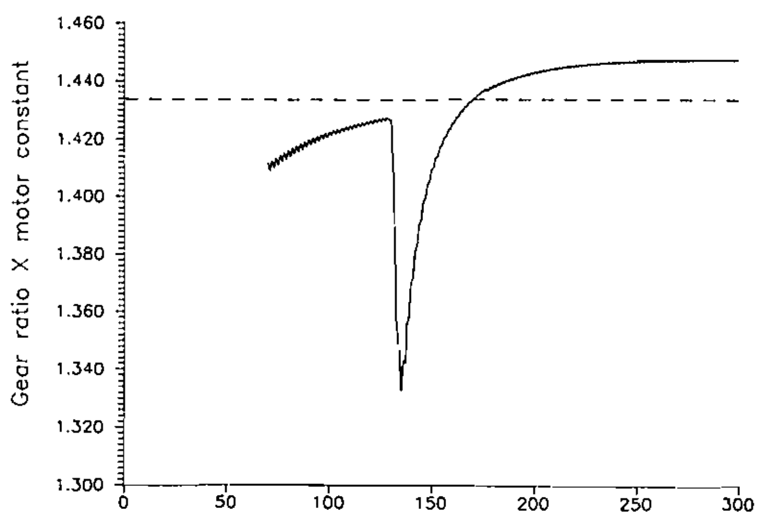
P3 (A)–(E) Motor characteristics, fault at $t = 130$, $ff = (1 \cdot 1)$.



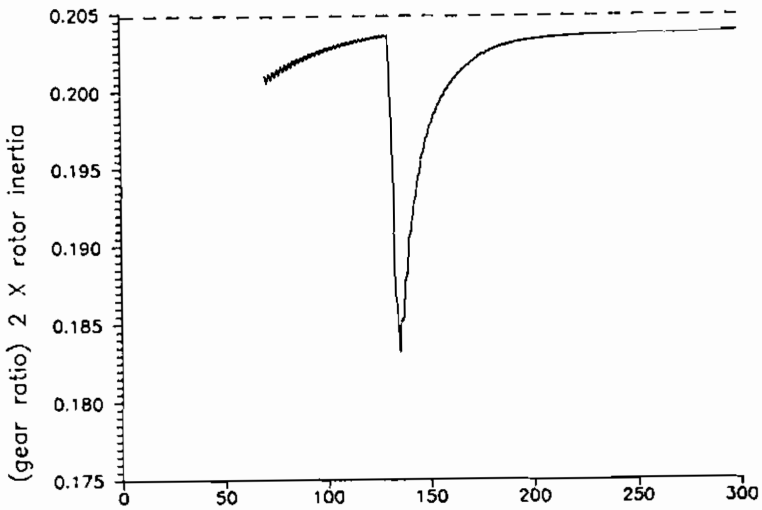
(A)


0.040
0.030
0.020
0.010
0.000
INDUCTANCE
0
50
100
150
200
250
300

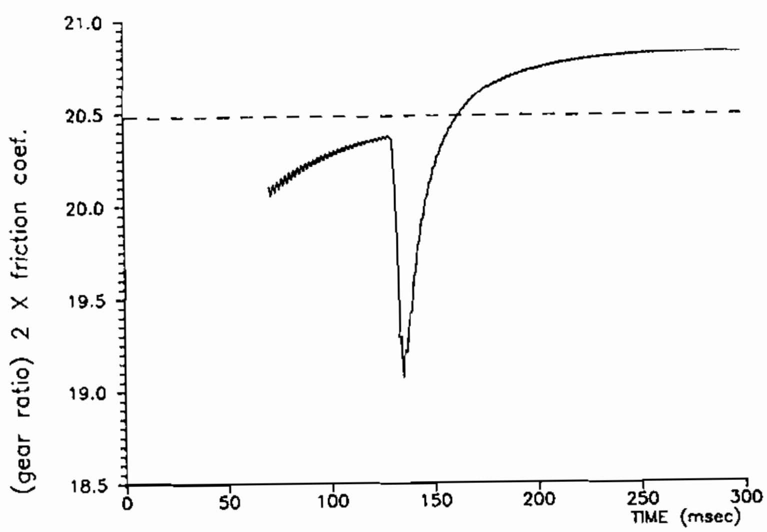
(*B*)


1.460
1.440
1.420
1.400
1.380
1.360
1.340
1.320
1.300
Gear ratio X motor constant
0
50
100
150
200
250
300

(*C*)

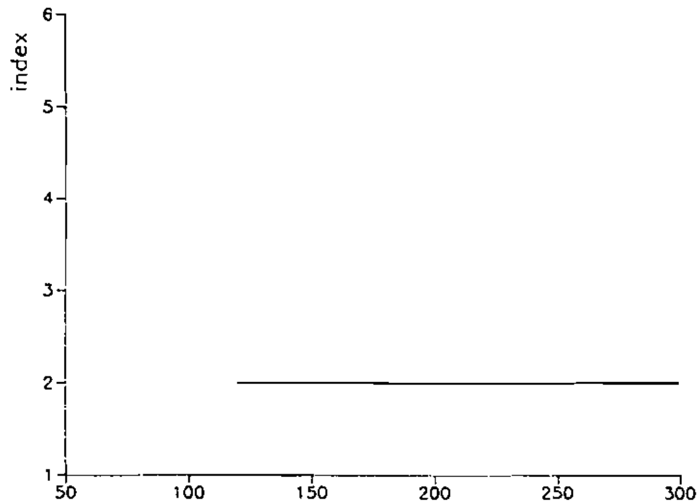


(*D*)

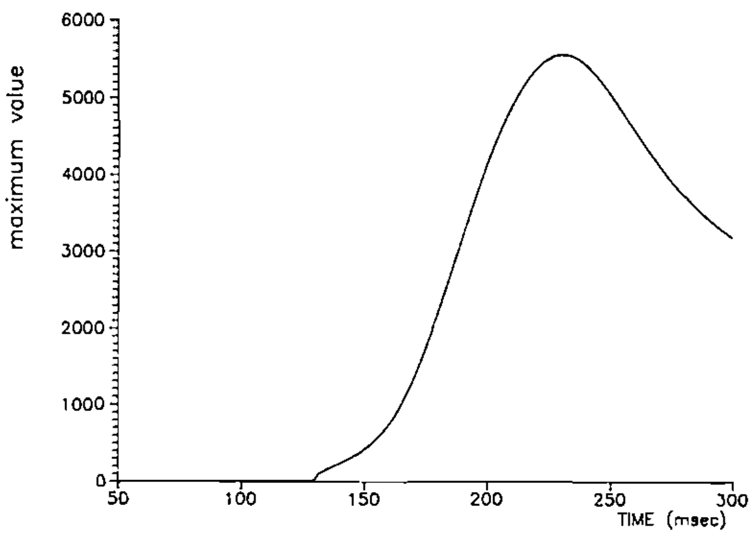


(*E*)

P4 (*A*)–(*E*) Motor characteristic fault at $t = 130, ff = (0{\cdot}95, 0{\cdot}99)$.

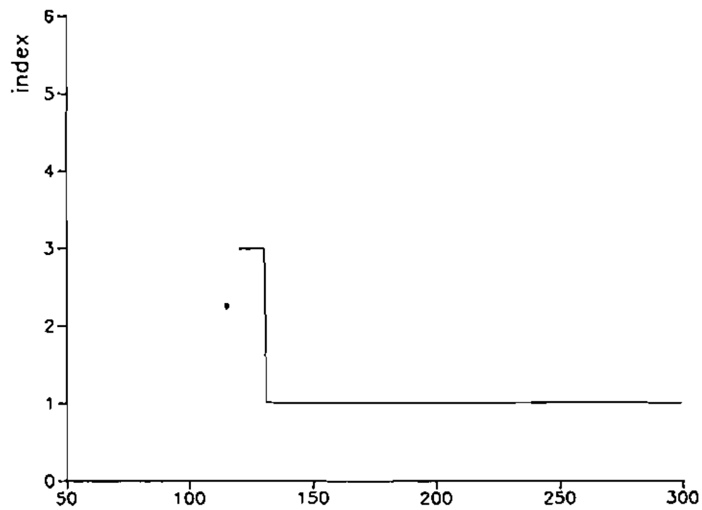


(*A*)

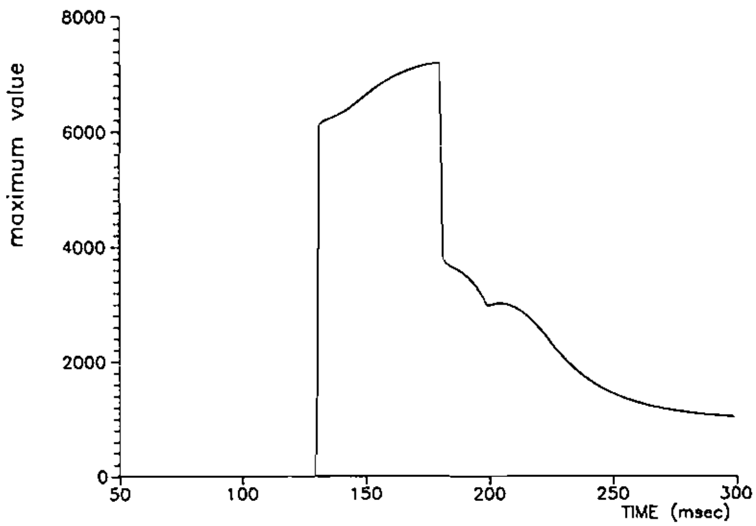


(*B*)

P5 (*A*), (*B*) Maximum value of likelihood function and its parameter index, fault at $t = 130$, $ff = (1 \cdot 1)$.



(A)



(B)

P6 (A), (B) Maximum value of likelihood function and its parameter index, fault at $t = 130$, $ff = (0{\cdot}95, 0{\cdot}99)$.

Figure 3.

In plots P4(A)–(E) the same estimator with a forgetting factor matrix $B=\text{diag}\{0{\cdot}95, 0{\cdot}95, 0{\cdot}95, 0{\cdot}99, 0{\cdot}99, 0{\cdot}99\}$ is used. The improvement is obvious. All estimates converge quite quickly to their respective true values. The exact estimated values are shown, for both cases, in Table 1.

	R	L	$K_m N$	$J_m N^2$	ρN^2
True value	1·04 (1·09)	0·00089	1·4336	0·2048	20·48
No forgetting factor	1·3743	0·011396	1·4424	0·2041	20·705
Forgetting factor	1·10	0·000896	1·4476	0·2038	20·821

Table 1. Estimate values using least-squares estimate after 300 samples.

In plots P5(A), (B) the detection procedure is shown in terms of the value of the likelihood functions of the various parameters. The two diagrams refer to the maximum value attained at each time instant by the likelihood functions and the parameter to which this maximum pertains. As expected, from the estimation results, the absence of the forgetting factor results in a wrong identification of the failed parameters, even though an alarm has correctly been sounded at $t = 131$. (The speed is rather surprising.) This error is corrected by the introduction of the forgetting factor, as seen in plots P6(A), (B).

The discrete square root method did not give satisfactory results in the case of fault, irrespective of the value of the forgetting factor (scalar). The results are summarized in Table 2. Further, the run-time of the second method was, in comparison, much greater than the first one. Even though absolute measurements have not much meaning, since the simulation tests were performed on an IBM-compatible and no special attempt was made to optimize the programming, the first method is to be preferred on grounds of speed.

	R	L	$K_m N$	$J_m N^2$	ρN^2
True value	1·04 (1·09)	0·00089	1·4336	0·2048	20·48
No-fault, no forgetting factor	1·0385	0·0009	1·4548	0·1645	21·02
No-fault, forgetting factor (0·95)	1·0385	0·0009	1·4544	0·0433	21·69
Fault, no forgetting factor	1·377	0·014	−6·154	−0·8683	−88·42
Fault, forgetting factor (0·95)	1·09	0·00089	1·433	−0·159	19·29

Table 2. Estimate values using DSF algorithms after 30 samples.

6. Conclusions

In view of the simulation results the following are concluded.

(i) A fault detection mechanism is needed for the particular type of robot/controller combination, if optimum performance is required at all times.

(ii) A detection mechanism based on maximum likelihood estimation with forgetting factor, making it essentially a moving window estimator, has been shown to work quite satisfactorily. Even though only simulation tests were performed, it is believed that due to the recursive nature of all the calculations, the proposed method is fast enough for on-line implementation.

(iii) Further work is necessary on an actual robot in order to fully test the proposed mechanism. Possibly reorganization following detection and estimation is also an area for further research.

Appendix A

The dynamic equations for the armature circuit and the mechanics of a dc motor lead to the state–space representation for the actuator of the link i of the robot

$$\begin{bmatrix} \dfrac{di_i}{dt} \\ \dfrac{d\omega_i}{dt} \end{bmatrix} = \begin{bmatrix} -\dfrac{R_i}{L_i} & -\dfrac{K_{mi}N_i}{L_i} \\ \dfrac{k_{mi}}{J_{mi}N_i} & -\dfrac{\rho_i}{J_{mi}} \end{bmatrix} \begin{bmatrix} i_i \\ \omega_i \end{bmatrix} + \begin{bmatrix} -\dfrac{1}{L_i} & 0 \\ 0 & -\dfrac{1}{J_{mi}N_i^2} \end{bmatrix} \begin{bmatrix} V_i(t) \\ T_{Li}(t) \end{bmatrix} \qquad \text{(A 1)}$$

where V_i is the calculated control voltage from the MRAC controller and T_{Li} is the corresponding torque generated at the ith joint.

Appendix B

By observing the MIMO-linear continuous-time system (4) over $k \geqslant l$ samples, the following discrete time set of k weighted equations is obtained

$$\left.\begin{aligned} \lambda^{k-1}e(1) &= \lambda^{k-1}y^{(n)}(1) - \lambda^{k-1}\Psi(1)\theta \\ \lambda^{k-2}e(2) &= \lambda^{k-2}y^{(n)}(2) - \lambda^{k-2}\Psi(2)\theta \\ &\vdots \\ \lambda^{0}e(k) &= \lambda^{0}y^{(n)}(k) - \lambda^{0}\Psi(k)\theta \end{aligned}\right\} \qquad \text{(B 1)}$$

The weighted squares norm to be minimized is given by (6)

$$V_e(\kappa) = \|E_e(k)\|^2 = \|D_e(k)\theta^*\|^2 \qquad \text{(B 2)}$$

where

$$\theta^* \equiv \begin{bmatrix} -\theta \\ 1 \end{bmatrix} \in R^{(rl+1)}, \quad D_e(k) \in R^{kr \times (rl+1)}$$

The basic idea of the discrete square root filtering applied here is as follows. Given

$$D_e(k)\theta^* = E_e(k)$$

find an orthogonal transformation $T(k) \in R^{kr \times kr}$ such that

$$T(k)D_e(k) = \begin{bmatrix} \overbrace{D_u(k)}^{rl+1} \\ \hline 0 \end{bmatrix} \begin{matrix} \}\, rl+1 \\ \}\, kr-(rl+1) \end{matrix}, \quad T(k)^{\mathrm{T}}T(k) = I, \quad k = l, l+1, \ldots \qquad \text{(B 3)}$$

where $D_u(k)$ is upper triangular, or equivalently, find $D_u(k)$ and $E' = (k) = T(k)E_e(k)$ directly.

This transformation does not change the cost function, i.e.

$$V_e(k) = \|E_e(k)\|^2 = \|T(k)E_e(k)\|^2 = \|E'_e(k)\|^2 \tag{B 4}$$

Equation (B 3) leads to the estimator of θ given by (10). For details see Geiger (1986) and Kaminski *et al.* (1971). Consider the data at the instant $k+1$. If the data weighting procedure defined by (6) and (6 *a*) is considered, the system observation (4) at the instant $k+1$ becomes

$$\lambda^{-1}e(k+1) = \lambda^{-1}y^{(n)}(k) - \lambda^{-1}\Psi(k)\theta \tag{B 5}$$

By multiplying both sides of (B 1) and (B 5) by λ, the following set of $k+1$ weighted equations is obtained

$$\left.\begin{aligned} \lambda^k e(1) &= \lambda^k y^{(n)}(1) - \lambda^k \Psi(1)\theta \\ \lambda^{k-1}e(2) &= \lambda^{k-1}y^{(n)}(2) - \lambda^{k-1}\Psi(2)\theta \\ &\vdots \qquad\qquad \vdots \\ \lambda e(k) &= \lambda y^{(n)}(k) - \lambda\Psi(k)\theta \\ e(k+1) &= y^{(n)}(k+1) - \Psi(k+1)\theta \end{aligned}\right\} \tag{B 6}$$

From (B 1), (B 3), (B 6), 6 and 8, it follows that

$$E_e(k+1) = \begin{bmatrix} \lambda E_e(k) \\ \hline z(k+1)\theta^* \end{bmatrix} = \begin{bmatrix} \lambda D_e(k) \\ \hline z(k+1) \end{bmatrix}\theta^* \equiv D_e(k+1)\theta^*$$

Proposition

Finding an orthogonal transformation $T(k+1) \in R^{(k+1)r \times (k+1)r}$, $k \geqslant l$, such that

$$T(k+1)D_e(\kappa+1) = \begin{bmatrix} D_u(k+1) \\ 0 \end{bmatrix} \tag{B 7}$$

is equivalent to finding an orthogonal transformation $T(k+1) \in R^{[r(l+1)+1]\times[r(l+1)+1]}$ such that

$$T(k+1)\begin{bmatrix} \lambda D_u(k) \\ z(\kappa+1) \end{bmatrix} = \begin{array}{c} \overbrace{\qquad\quad}^{rl+1} \\ \begin{bmatrix} D_u(k+1) \\ 0 \end{bmatrix} \end{array} \begin{array}{l} \}\, rl+1 \\ \}\, r \end{array} \tag{B 8}$$

where $z(k+1)$ is the $(k+1)$th group of data matrix, defined by (8).

Proof

The Householder and MGS triangularization algorithms have the property that given $A \in R^{(n+r)\times n}$, then

$$TA = \begin{array}{c} \overbrace{\quad}^{n} \\ \begin{bmatrix} w \\ 0 \end{bmatrix} \end{array} \begin{array}{l} \}\, n \\ \}\, r \end{array} \Leftrightarrow T(\lambda A) = \begin{bmatrix} (\lambda w) \\ 0 \end{bmatrix} \begin{array}{l} \}\, n \\ \}\, r \end{array}, \quad \lambda \in R \quad \text{and} \quad 0 < \lambda < 1 \tag{B 9}$$

where T is an orthogonal transformation. In addition, from (B 8), using (B 7), and the property (B 9) it follows that

$$\left\| T(k+1)\begin{bmatrix} \lambda D_u(k) \\ \hdashline z(k+1) \end{bmatrix}\theta^* \right\|^2 = \left\| T(k+1)\begin{bmatrix} \lambda D_u(k) \\ \vdots \\ 0 \\ \vdots \\ z(k+1) \end{bmatrix}\theta^* \right\|^2$$

$$= \left\| T(k+1)\begin{bmatrix} T(K)\lambda D_e(k) \\ \hdashline z(k+1) \end{bmatrix}\theta^* \right\|^2$$

$$= \theta^{*\mathrm{T}}[\lambda D_e(k)^{\mathrm{T}} T(k)^{\mathrm{T}} \quad z(k+1)^{\mathrm{T}}] T(k+1)^{\mathrm{T}} T(k+1)\begin{bmatrix} T(k)\lambda D_e(k) \\ \hdashline z(k+1) \end{bmatrix}\theta^*$$

$$= \theta^{*\mathrm{T}}[\lambda^2 D_e(k)^{\mathrm{T}} T(k)^{\mathrm{T}} T(k) D_e(k) + z^{\mathrm{T}}(k+1)z(k+1)]\theta^*$$

$$= \theta^{*\mathrm{T}}[\lambda D_e(k)^{\mathrm{T}} \quad z(k+1)^{\mathrm{T}}]$$

$$\times \begin{bmatrix} \lambda D_e(k) \\ z(\kappa+1) \end{bmatrix}\theta^* = \left\| \begin{bmatrix} \lambda D_e(k) \\ \hdashline z(k+1) \end{bmatrix}\theta^* \right\|^2 = \|D_e(k+1)\theta^*\|^2 = V_e(k+1)$$

Thus the D-DSF algorithm of Lemma 3.2 can be derived. This implies also that the data matrix to be 'compressed' by the triangularization algorithm at any instant $k > 1$ has constant dimension, depending on the number of parameters to be estimated and the order r of the physical process vector $y(t)$ and it is independent of the sampling instant k. □

Appendix C

The quantity $\tilde{\sigma}_i^2(k)$ is given by

$$\tilde{\sigma}_i^2(k) = \frac{1}{N_w}\sum_{j=k-N_w+1}^{k} [\hat{p}_i(j) - \bar{\mu}_i]^2$$

This is the same as

$$\tilde{\sigma}_i^2(k) = \frac{1}{N_w}\left[\sum_{j=k-N_w}^{k-1} (\hat{p}_i(j) - \bar{\mu}_i)^2 - (\hat{p}_i(k-N_w) - \bar{\mu}_i)^2 + (\hat{p}_i(k) - \bar{\mu}_i)^2\right]$$

$$\subset \tilde{\sigma}_i^2(k-1) - \frac{1}{N_w}[\hat{p}_i^2(k-N_w) - \hat{p}_i^2(k) - 2\bar{\mu}_i(\hat{p}_i(k-N_w) - \hat{p}_i(k))]$$

$$= \tilde{\sigma}_i^2(k-1) - \frac{1}{N_w}[\hat{p}_i^2(k-N_w) - \hat{p}_i^2(k) - 2\bar{\mu}_i\gamma_i(k)]$$

where $\gamma_i(k) = \hat{p}_i(k-N_w) - \hat{p}_i(k)$.

REFERENCES

GEIGER, G., 1986, *Int. Modelling Simulation*, **6**, 26.

ISERMANN, R., 1982, *Automatica*, **18**, 513; 1984, *Ibid.*, **20**, 387.

Kaminski, P., Bryson, A., and Schmidt, S., 1971, *I.E.E.E. Trans. autom. Control*, **16**, 727.

Nicosia, S., and Tomei, P., 1984, *Automatica*, **20**, 635.

Paul, R. P., 1981, *Robotic Manipulators: Mathematics, Programming and Control* (Cambridge, MA: MIT Press).

Pouliezos, A., 1980, *I.E.E.E. Trans. autom. Control*, **25**, 834.

Pouliezos, A., Tzafestas, S., and Stavrakakis, G., 1986, Fault detection and location in MRAC Robotic System. *Proc. of 1st European Workshop on Fault Diagnostics, Reliability and Related Knowledged-Based Approaches*, Island of Rhodes, Greece.

Tzafestas, S., 1985, Reliability and fault detection techniques of large scale systems, *Real Time Control of Large Scale Systems*, edited by G. Schmidt, M. Singh, A. Titli, and S. Tzafestas (Berlin: Springer-Verlag).

Tzafestas, S., and Stavrakakis, G., 1986, Model reference adaptive control of industrial robots with actuator dynamics. *IFAC/IFIP/IMACS Int. Symp. on Theory of Robots*, Vienna.