

Threshold Selection Based on Closed-loop Performance for Fault Detection Schemes

S. Aberkane* M. Kinnaert**

* CRAN – CNRS UMR 7039, Université Henri Poincaré, Nancy 1, BP 239,

F-54506 Vandœuvre-lès-Nancy Cedex

e-mail: samir.aberkane@cran.uhp-nancy.fr

** Control Engineering Department, CP 165/55, Université libre de Bruxelles (ULB)

50 Av. F.D. Roosevelt, B-1050 Brussels, BELGIUM

e-mail: michel.kinnaert@ulb.ac.be

Abstract: The threshold selection issue is considered in a fault detection system which is a part of an active fault tolerant control scheme. The proposed approach takes explicitly into account the closed-loop performance of the regulated system. Indeed, the threshold is selected to ensure optimal regulation performance over a specified time horizon, while taking into account possible malfunctions affecting the system. More specifically sensor faults, like bias or drift, are considered in combination with a reconfiguration scheme made of software sensors. A numerical algorithm based on randomization techniques is provided and its running is illustrated on a numerical example of FTC for a winding machine.

1. INTRODUCTION

Fault tolerant control (FTC) systems aim at guaranteeing that a process or device keeps fulfilling its mission even in the presence of faults, although possibly in a degraded mode. It involves automatically detecting and isolating the faulty components thanks to an appropriate diagnosis systems, and then reconfiguring the control law on-line in response to the information provided by the diagnosis system Blanke et al. [2006]. Despite the evident interaction between fault detection and isolation (FDI) and reconfiguration algorithms, the research on FDI and reconfiguration methods has often evolved separately, certainly because of the difficulty of each of these problems. Typically, in the reconfiguration literature, it is assumed that a perfect FDI device is available (*i.e.* no detection delay, no false alarm, no missed detection ...*etc.*) Gao et al. [1991], Lunze et al. [2006]. On the other hand, the figures of merit used for FDI algorithms are the detection delay and false alarm rate, and an algorithm is judged satisfactory if it detects failures quickly and generates few false alarms, regardless of the performance of the reconfigured closed loop system Basseville et al. [1993].

Some results addressing this interaction issue have appeared recently. For instance Gani et al. [2007] presents an approach that combines a fault detection filter, to detect actuator faults, and a reconfiguration policy based on stability region. The latter allows to switch to an appropriate controller upon occurrence of a fault. However, the design of the threshold in the detection system is not linked to closed-loop performance in that paper. This is probably due to the fact that total actuator failures are considered, although it is not explicitly mentioned in the text, so that sufficiently fast detection is always achieved and reconfiguration is compulsory. Other approaches rely on estimates of loss of control effectiveness of the actuators or estimates

of the bias of the sensors to either retune the control law Wu et al. [2000] or compensate for the bias Edwards et al. [2006]. However, since the fault estimate is never totally accurate, its use in a compensation scheme under healthy conditions actually deteriorates closed-loop performance, and it is thus important to complement such approaches with an appropriate decision scheme to determine when the fault estimate should be used for compensation as indicated in Edwards et al. [2006]. This goes in the direction of the research reported in the present paper.

Indeed, our aim is to address a question involving the interplay between FDI systems and reconfiguration mechanisms, namely the tuning of the threshold of the FDI systems based on the closed-loop performance of the FTC system. To keep the setting sufficiently simple, a system subject to sensor faults is considered, and the reconfiguration mechanism consists in substituting a software sensor for a faulty hardware sensor. Typically, closed-loop performance will be better with the hardware sensor than with the software sensor, however, the presence of a fault on the hardware sensor will also degrade the closed-loop performance. One should thus determine at what fault level switching to the software sensor is recommended in order to achieve the best closed-loop performance over a given mission time. The latter may correspond to the time interval between two maintenance operations for instance. Setting the fault level actually amounts to setting the threshold for the FDI algorithm which triggers the reconfiguration mechanism.

This paper is organized as follows. The fault tolerant (FTC) system is described in section 2. The threshold selection problem is stated in section 3, and an algorithm to solve it is presented. Finally, a winding machine example is used in section 4 to illustrate the approach.

2. DESCRIPTION OF THE FTC SYSTEM

2.1 System modeling

Consider the following class of discrete-time dynamical systems:

$$\varphi : \begin{cases} x_{k+1} = A_x x_k + B_u u_k + B_w w_k \\ y_k = C_y x_k + \nu_k + E_f f_k \\ z_k = C_z x_k + D_z u_k \end{cases} \quad (1)$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^r$ is the input, $y_k \in \mathbb{R}^q$ is the measured output, w_k is the process noise, ν_k is the measurement noise, z_k is the controlled output, and f_k represents sensor faults. The sequence $\{w_k\}_k$ is white and Gaussian ($w_k \sim \mathcal{N}(0, R_w)$ with R_w a symmetric positive definite matrix). ν_k is the output of the linear system

$$\varphi_\nu : \nu_{k+1} = A_\nu \nu_k + \eta_k \quad (2)$$

where $\{\eta_k\}_k$ is a Gaussian white noise sequence ($\eta_k \sim \mathcal{N}(0, R_\eta)$ with $R_\eta > 0$) independent of $\{w_k\}_k$. Thus ν_k is sequentially correlated. Model (2) corresponds to a band limited noise. Such a noise model is needed to ensure the existence of a solution to the \mathcal{H}_2 control problem stated in section 2.2. Besides, the initial state x_0 is assumed to be a Gaussian random vector with zero mean and variance Σ_0 , and similarly ν_0 has zero mean and variance Σ_{ν_0} .

A parameterized class of functions will be considered for f_k :

$$\begin{cases} f_k = 0 & \text{if } k < k_0 \\ f_k = f(k; k_0, \beta) & \text{otherwise} \end{cases} \quad (3)$$

with k_0 , the fault occurrence time, and β a parameter linked to fault magnitude. Both parameters are assumed to be random variables. The function $f(k; k_0, \beta)$ is also assumed to be linear w.r.t. parameter β . It is further supposed that the probability mass function (pmf), $p_{k_0}(k_0)$ of k_0 over the mission time is known, including the probability that no fault occurs during the mission, p_{nf} , and that the probability density function $p_\beta(\beta)$ of β is also known. β and k_0 are supposed to be independent.

2.2 Controller synthesis

In the absence of fault ($f_k = 0$), process (1) is controlled by static output feedback. Upon occurrence of a fault, the faulty sensor will be replaced by a software sensor that relies on the non faulty sensors. In the following, one single sensor fault is considered for the sake of simplicity. The nominal and the reconfigured control laws are successively presented in the next two subsections.

Nominal case The nominal controller is based on the model of the fault free system (1), and it has the form:

$$\varphi_s : u_k = \mathcal{K} y_k \quad (4)$$

Combining (1)-(2) we obtain the following dynamical model:

$$\begin{cases} \chi_{k+1} = \bar{A}_\chi \chi_k + \bar{B}_u u_k + \bar{E}_w w_k + \bar{E}_\eta \eta_k \\ y_k = \bar{C}_y \chi_k \\ z_k = \bar{C}_z \chi_k + D_z u_k \end{cases} \quad (5)$$

where $\chi_k = [x'_k \quad \nu'_k]'$ and

$$\bar{A}_\chi = \begin{bmatrix} A_x & \mathbf{0} \\ \mathbf{0} & A_\nu \end{bmatrix}, \bar{B}_u = \begin{bmatrix} B_u \\ \mathbf{0} \end{bmatrix}, \bar{E}_w = \begin{bmatrix} B_w \\ \mathbf{0} \end{bmatrix}, \\ \bar{E}_\eta = \begin{bmatrix} \mathbf{0} \\ \mathbb{I} \end{bmatrix}, \bar{C}_y = [C_y \ \mathbb{I}], \bar{C}_z = [C_z \ \mathbf{0}].$$

Our goal is to compute a static output feedback controller φ_s that stabilizes the closed loop system and ensures \mathcal{H}_2 performance, as specified below.

Applying the controller φ_s to (5), we obtain the following closed loop system:

$$\varphi_{cl} \begin{cases} \chi_{k+1} = \tilde{A}_\chi \chi_k + \tilde{E} \bar{w}_k \\ y_k = \tilde{C}_y \chi_k \\ z_k = \tilde{C}_z \chi_k \end{cases} \quad (6)$$

where

$$\bar{w}_k = [w'_k \quad \eta'_k]', \tilde{A}_\chi = \bar{A}_\chi + \bar{B}_u \mathcal{K} \bar{C}_y, \\ \tilde{E} = [\bar{E}_w \quad \bar{E}_\eta], \tilde{C}_z = \bar{C}_z + D_z \mathcal{K} \bar{C}_y.$$

Letting $T_{\bar{w}z}$ denote the discrete transfer matrix between \bar{w} and z , the \mathcal{H}_2 control problem can be stated as follows. Determine a stabilizing gain \mathcal{K} such that $\|T_{\bar{w}z}\|_2$ is minimum.

This can be achieved by finding the minimum value of γ_2 for which there exist a positive definite matrix \mathcal{P}_2 and a matrix \mathcal{K} such that

$$\text{tr}(\tilde{E}' \mathcal{P}_2 \tilde{E}) < \gamma_2^2$$

$$\tilde{A}'_\chi \mathcal{P}_2 \tilde{A}_\chi - \mathcal{P}_2 + \tilde{C}'_z \tilde{C}_z < 0$$

Then \mathcal{K} is a stabilizing gain such that $\|T_{\bar{w}z}\|_2 < \gamma_2$.

For synthesis purpose we use the LMI methods based on cone complementary techniques described in El Ghaoui et al. [1997].

Reconfiguration Let us consider, without loss of generality, that sensor 1 is faulty, then the proposed reconfigured control law is given by

$$u_{rk} = \mathcal{K}_1 \hat{y}_{1k} + \mathcal{K}_2 \tilde{y}_k \quad (7)$$

with $\mathcal{K} = [\mathcal{K}_1 \quad \mathcal{K}_2]$, $\tilde{y}_k = [y_{2k} \dots y_{qk}]'$ where $\tilde{y}_k = M y_k$ are the "healthy" outputs, and \hat{y}_{1k} is an estimation of y_{1k} obtained from the state estimate of the following Kalman filter (based on model (5) with measured output \tilde{y}_k):

$$\varphi^F : \begin{cases} \hat{\chi}_{k+1} = \bar{A}_\chi \hat{\chi}_k + \bar{B}_u u_{rk} + \mathcal{K}_{fr} (\tilde{y}_k - \hat{\tilde{y}}_k) \\ \hat{\tilde{y}}_k = M \bar{C}_y \hat{\chi}_k \end{cases} \quad (8)$$

where $\hat{\tilde{y}}_k$ is the estimation of \tilde{y}_k . This yields

$$\hat{y}_{1k} = M_1 \bar{C}_y \hat{\chi}_k. \quad (9)$$

where M_1 simply selects the first component of vector $\bar{C}_y \hat{\chi}_k$. In accordance with the standard Kalman filter theory, the initial state $\hat{\chi}_0$ is assumed to be a Gaussian random vector with zero mean and variance $\hat{\Sigma}_0 = \text{diag}(\Sigma_0 \quad \Sigma_{\nu_0})$.

Note that the filter (8) only uses the healthy outputs \tilde{y}_k to provide robust estimation (robustness w.r.t sensor faults) of the faulty output. When writing (8), it is implicitly assumed that the required detectability and stabilizability conditions are verified to be able to compute a unique stabilizing Kalman filter gain \mathcal{K}_{fr} .

Remark: In order to avoid a transient due to the initialization of the filter (8) upon reconfiguration, this filter is running both in faulty and in fault free conditions. In the latter case, u_k as given by (4) is substituted for u_{rk} .

In the next section, the fault detection system that decides when to switch from control law (4) to control law (7)-(9) is described.

2.3 Fault detection (FD)

The two parts that constitute the FD system, namely the residual generator and the decision module are successively described. The first generates fault indicators, called residuals, which have zero mean in the absence of fault and nonzero mean upon occurrence of a fault. The decision module then performs a test of hypothesis to determine the possible presence of a fault.

Residual generation A Kalman filter for system (1)-(2) will be used as a residual generator. However, for a system made of the concatenation of equations (1) and (2), the Kalman filtering problem is singular Jazwinski [1970], pages 212-215. One way to solve such a problem is to consider the following modification of the measurement equation.

From (1), (2) with $f_k = 0$, we have

$$\begin{aligned} y_k - A_\nu y_{k-1} &= [C_y A_x - A_\nu C_y] x_{k-1} + C_y B_u u_{k-1} \\ &\quad + C_y B_w w_{k-1} + \eta_{k-1} \\ &= \hat{C}_y x_{k-1} + \hat{B}_u u_{k-1} + \hat{B}_w w_{k-1} + \eta_{k-1} \end{aligned} \quad (10)$$

Combining (10) with (1) gives the following state space representation with modified output:

$$\varphi_{mod} : \begin{cases} x_{k+1} = A_x x_k + B_u u_k + B_w w_k \\ \bar{y}_k = (y_{k+1} - A_\nu y_k) = \hat{C}_y x_k + \hat{B}_u u_k + \hat{B}_w w_k + \eta_k \end{cases} \quad (11)$$

in which the measurement noise variance is now nonsingular.

Under standard detectability and stabilizability conditions, a steady state Kalman filter for system (11) can be determined. It takes the form

$$\varphi_{FD} : \begin{cases} \hat{x}_{k+1} = A_x \hat{x}_k + B_u u_k + \mathcal{K}_f (\bar{y}_k - \hat{y}_k) \\ \hat{y}_k = \hat{C}_y \hat{x}_k + \hat{B}_u u_k \end{cases} \quad (12)$$

where \hat{x}_0 is a Gaussian random vector with zero mean and variance $\hat{\Sigma}_{FD} = \Sigma_0$. Let us define $e_k = x_k - \hat{x}_k$, then (11), (12) yield

$$e_{k+1} = \check{A} e_k + \check{B} w_k - \mathcal{K}_f \eta_k \quad (13)$$

where

$$\check{A} = A_x - \mathcal{K}_f \hat{C}_y, \quad \check{B} = B_w - \mathcal{K}_f \hat{B}_w. \quad (14)$$

The residual r_k is the innovation of the Kalman filter:

$$r_k = \bar{y}_k - \hat{y}_k = \hat{C}_y e_k + \hat{B}_w w_k + \eta_k \quad (15)$$

Decision module Given the Gaussian assumption for the noise sequences $\{w_k\}_k$ and $\{\eta_k\}_k$, r_k is normally distributed with zero mean in the absence fault. A fault normally induces a change in the mean of the residual Blanke et al. [2006], chapter 6. Therefore, considering the vector $R_k = [r'_k \ r'_{k-1} \ \dots \ r'_{k-l}]'$, obtained by stacking up $(l+1)$ residual samples, the residual evaluation problem can be stated as the following hypothesis testing problem: Given a realization of the random vector R_k , choose between the following two hypotheses

$$\mathcal{H}_0: \mathcal{L}(R_k) = \mathcal{N}(0, Q)$$

$$\mathcal{H}_1: \mathcal{L}(R_k) = \mathcal{N}(\mu_k, Q) \text{ where } \mu_k \text{ is an unknown non-zero vector.}$$

A standard χ^2 -test is used to solve this problem, based on the decision function $\xi_k = R'_k \mathbf{Q}^{-1} R_k$ which has a χ^2 distribution with non centrality parameter equal to zero under hypothesis \mathcal{H}_0 . Specifically,

$$\begin{cases} \text{if } \xi_k < h, \text{ decide } \mathcal{H}_0, \\ \text{otherwise, decide } \mathcal{H}_1, \end{cases} \quad (16)$$

where the threshold h is a tuning parameter.

The variance of R_k , denoted \mathbf{Q} , can be computed as indicated in Aberkane et al. [2008].

In typical FD systems, the threshold h used for the χ^2 -test is adjusted in such a way that, in the absence of fault,

$$\text{Prob}(\xi_k > h) = \alpha \quad (17)$$

where α is a specified false alarm (FA) probability. However, in a fault tolerant control context, it is not clear what value to choose for α . A low FA probability would a priori be beneficial; but it implies a large value for h , and hence a large value for the smallest detectable fault. This is the classical trade off between false alarm and missed detection. In FTC, what really matters is closed-loop performance. Therefore, in the next section an optimization problem is stated, for the determination of h , on the basis of this criterion.

3. THRESHOLD OPTIMIZATION

Consider system (1) subject to fault f_k described by the specific family of functions (3). Let the mission time (for instance the time interval between two sensor maintenance operations) be equal to H , and assume, for the sake of simplicity, that a single switching from control law (4) to control law (7)-(9) can take place. Such a switching is triggered by the FD system either upon occurrence of a fault or due to a false alarm. Our aim is to determine the threshold h of test (16) in such a way that a closed-loop performance measure, averaged over all possible faults within family (3), over all possible noise realizations and all possible reactions of the FD system (accounting for the false alarms and detection delay), is minimized.

A mathematical expression for this optimization problem is:

$$\min_h \mathcal{E} \left(\sum_{k=0}^H z'_k z_k \right) = \min_h \mathcal{J} \quad (18)$$

where the expectation operator \mathcal{E} carries over the noise sequences, the initial state and the fault sequence.

The difficulty in the evaluation of this cost function stems from the decision system which nonlinear function and threshold impose switching between the nominal and the reconfigured control law at a random time instant that depends on the realizations of the different random vectors and processes appearing in the problem statement. A simple way to evaluate the cost would be to resort to standard Monte Carlo simulation. However, this would not exploit the linearity property of the closed-loop dynamical system both in fault free and in faulty operation. The algorithm presented next attempts to combine Monte Carlo simulation and classical results on stochastic linear dynamical systems to limit the computation time of the cost function \mathcal{J} for a given threshold h .

The contributions of the fault free trajectories and the trajectories subject to fault occurrence are successively considered. Next, the way to combine these results in order to evaluate \mathcal{J} is explained.

3.1 Contributions of the fault free trajectories

A two step approach is used. First Monte Carlo simulation allows to determine, at each time instant, the mean and variance of the trajectories that yield no false alarm or that yield a false alarm. Next the contribution to the cost after a false alarm is triggered is computed.

The following algorithm is used for Monte Carlo simulation:

- Obtain N realizations of $\chi_0, \hat{\chi}_0, \hat{x}_0$, say $\chi_0^n, \hat{\chi}_0^n, \hat{x}_0^n, n = 1, \dots, N$.

- For i ranging from 1 to H

Step 1 Obtain N realizations of w_{i-1} and η_{i-1} , say $w_{i-1}^n, \eta_{i-1}^n, n = 1, \dots, N$.

Step 2 Perform, for each n , one simulation step of the dynamic system (4),(5), (8), (13), (15), and evaluate the associated decision variable ξ_i^n .

Step 3 Let $\xi_i^{n_j}, j = 1, \dots, N_i$ ($j = N_i+1, \dots, N_i + \tilde{N}_i$) denote the set of values of the decision variable that cross (are below) the threshold h at time instant i , and let $\chi_{nom}^{n_j}(i) = ((\chi_i^{n_j})' (\hat{\chi}_i^{n_j})' (e_i^{n_j})')'$ be the associated state of the concatenated dynamical system (4), (5), (8), (13), then the mean and variance of the distribution of the state that yields a false alarm at time instant i is evaluated as

$$\begin{cases} m_{nom}^{fa}(i) = \frac{1}{N_i} \sum_{j=1}^{N_i} \chi_{nom}^{n_j}(i) \\ Q_{nom}^{fa}(i) = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (\chi_{nom}^{n_j}(i) - m_{nom}^{fa}(i)) \star \end{cases} \quad (19)$$

where the index "nom" indicates that the nominal control law is used, and the symbol \star indicates the transpose element, i.e., $X \star = X X'$, and X is any given matrix.

The mean and variance of the distribution of the state that yields no false alarm at time instant i , $m_{nom}^{nfa}(i)$ and $Q_{nom}^{nfa}(i)$ are estimated in a similar way as (19) from the \tilde{N}_i relevant trajectories.

The mean and variance of z_i , upon occurrence of a false alarm, $m_{nom,z}^{fa}(i)$, $Q_{nom,z}^{fa}(i)$ or in the absence of false alarm, $m_{nom,z}^{nfa}(i)$, $Q_{nom,z}^{nfa}(i)$ are deduced from

$$z_i = \mathcal{C}_{nom} \chi_{nom}(i) \quad (20)$$

with $\mathcal{C}_{nom} = (\bar{C}_z + D_z \mathcal{K} \bar{C}_y \quad \mathbf{0} \quad \mathbf{0})$

Once a false alarm occurs, say at time k_{fa} , the controller is automatically reconfigured, and the closed-loop dynamics is described by equations (5), (7),(8). The mean, $m_{rec}(i; k_{fa})$ and variance $Q_{rec}(i; k_{fa})$ of the state of the reconfigured system, $\chi_{rec}(i) = (\chi_i' \hat{\chi}_i)'$ can be computed from the following equations.

$$\begin{cases} m_{rec}(i+1; k_{fa}) = \mathcal{A}_{rec} m_{rec}(i; k_{fa}) \\ Q_{rec}(i+1; k_{fa}) = \mathcal{A}_{rec} Q_{rec}(i; k_{fa}) \mathcal{A}'_{rec} \\ \quad + \mathcal{B}_{rec} \begin{pmatrix} R_w & \mathbf{0} \\ \mathbf{0} & R_v \end{pmatrix} \mathcal{B}'_{rec} \end{cases} \quad (21)$$

with initial conditions determined from (19) as

$$m_{rec}(k_{fa}; k_{fa}) = \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} \end{pmatrix} m_{nom}^{fa}(k_{fa})$$

and

$$Q_{rec}(k_{fa}; k_{fa}) = \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} \end{pmatrix} Q_{nom}^{fa}(k_{fa}) \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} \end{pmatrix}'$$

where

$$\mathcal{A}_{rec} = \begin{bmatrix} (\bar{A}_x + \bar{B}_u \mathcal{K}_2 M \bar{C}_y) & \bar{B}_u \mathcal{K}_1 M_1 \bar{C}_y \\ (\bar{B}_u \mathcal{K}_2 M \bar{C}_y + \mathcal{K}_{rf} M \bar{C}_y) & (\bar{A}_x - \mathcal{K}_{rf} M \bar{C}_y + \bar{B}_u \mathcal{K}_1 M_1 \bar{C}_y) \end{bmatrix}$$

$$\mathcal{B}_{rec} = \begin{bmatrix} \bar{E}_w & \bar{E}_\eta \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

characterize the closed-loop system.

The resulting mean and variance of z are deduced from

$$\begin{cases} m_{rec,z}(i; k_{fa}) = \mathcal{C}_{rec} m_{rec}(i; k_{fa}) \\ Q_{rec,z}(i; k_{fa}) = \mathcal{C}_{rec} Q_{rec}(i; k_{fa}) \mathcal{C}'_{rec} \end{cases} \quad (22)$$

with $\mathcal{C}_{rec} = [(\bar{C}_z + D_z \mathcal{K}_2 M \bar{C}_y \quad D_z) \quad \mathcal{K}_1 M \bar{C}_y]$.

3.2 Contribution of the faulty trajectory

Let us assume that a fault occurs at time k_0 , in the interval $[1, H]$. Considering the same N trajectories as in the fault free case, there will be $\sum_{\ell=1}^{k_0-1} N_\ell$ trajectories for which a false alarm will arise in the time interval $[1, k_0 - 1]$. Since reconfiguration will take place before fault occurrence, the contribution of these trajectories to the cost function can be computed using the same approach as above. For the remaining \tilde{N}_{k_0-1} trajectories, the fault occurring at time k_0 is possibly detected at some time $k_{fd} > k_0$. A reconfiguration then takes place at time k_{fd} . Small faults are not necessarily detected, which can be associated to a situation where $k_{fd} > H$. The detection delay is a stochastic variable which distribution can only be estimated by Monte Carlo simulation in the present context. However, in the study of on-line change detection algorithms, one often characterizes performance in terms of the *average run length* (ARL) function Basseville et al. [1993], namely the average time the algorithm runs before an alarm is generated (in the fault free case) and the mean time for detection (in the faulty case). The ARL depends upon the initial value of the decision function and the fault magnitude. It has been studied for the EWMA chart and the CUSUM algorithm in Wieringa [1999] and Basseville et al. [1993], and its computation is quite involved. In this paper, we resort to a simpler decision method for which the ARL function upon occurrence of fault with magnitude β will be estimated from the mean of the decision function. Letting $\widehat{ARL}_h(\beta)$ denote the estimated ARL function for a decision system with threshold h , one can write:

$$\widehat{ARL}_h(\beta) = \mathcal{E}(k_{fd} - k_0) \simeq k_{fd}^m - k_0 \quad (23)$$

where k_{fd}^m is the time instant at which the mean of the decision function becomes greater than the threshold, namely:

$$k_{fd}^m = \min_k (\mathcal{E}(R'_k \mathbf{Q}^{-1} R_k) > h) \quad (24)$$

Here $\mathcal{E}(R'_k \mathbf{Q}^{-1} R_k)$ is computed from:

$$\mathcal{E}(R'_k \mathbf{Q}^{-1} R_k) = \text{tr}(\Sigma_R(k) \mathbf{Q}^{-1}) + m'_R(k) \mathbf{Q}^{-1} m_R(k)$$

where $m_R(k)$ and $\Sigma_R(k)$ are respectively the mean and the variance of R_k that can be computed as detailed in Aberkane et al. [2008].

For a fault at time k_0 with magnitude β , we shall approximate the trajectory $z_k, k > k_0$ by assuming that the system operates with the faulty sensor and the nominal control law from time k_0 to time $k_0 + \widehat{ARL}_h(\beta) - 1$, and that reconfiguration takes place at time $k_0 + \widehat{ARL}_h(\beta)$.

In the time interval $[k_0, k_0 + \widehat{ARL}_h(\beta) - 1]$, the contribution $\mathcal{E}(z'_k z_k)$ associated to such trajectories, for a given fault magnitude β , takes the form:

$$\mathcal{E}(z'_k z_k) = \text{tr} \left(Q_{nom,\beta}(k; k_0) C_{nom} C'_{nom} + C_{nom} m_{nom,\beta}(k; k_0) m_{nom,\beta}(k; k_0)' C'_{nom} \right) \quad (25)$$

where $m_{nom,\beta}(k; k_0)$ and $Q_{nom,\beta}(k; k_0)$ are computed from

$$\begin{cases} m_{nom,\beta}(k+1; k_0) = \mathcal{A}_{nom} m_{nom,\beta}(k; k_0) + \mathcal{F}_{nom} f_k \\ Q_{nom}(k+1; k_0) = \mathcal{A}_{nom} Q_{nom}(k; k_0) \mathcal{A}'_{nom} + \mathcal{B}_{nom} \begin{pmatrix} R_w & 0 \\ 0 & R_\eta \end{pmatrix} \mathcal{B}'_{nom} \\ k_0 - 1 \leq k \leq k_0 + \widehat{ARL}_h(\beta) - 1 \end{cases} \quad (26)$$

with

$$\begin{aligned} \mathcal{A}_{nom} &= \begin{bmatrix} (\bar{A}_x + \bar{B}_u \mathcal{K} \bar{C}_y) & \mathbf{0} \\ (\bar{B}_u \mathcal{K} \bar{C}_y + \mathcal{K}_{fr} M \bar{C}_y) & (\bar{A}_x - \mathcal{K}_{fr} M \bar{C}_y) \end{bmatrix}, \\ \mathcal{B}_{nom} &= \begin{bmatrix} \bar{E}_w & \bar{E}_\eta \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathcal{C}_{nom} = \begin{bmatrix} \bar{C}_z + D_z \mathcal{K} \bar{C}_y & \mathbf{0} \end{bmatrix}, \\ \mathcal{F}_{nom} &= \begin{bmatrix} \bar{B}_u \mathcal{K} E_f \\ \bar{B}_u \mathcal{K} E_f + \mathcal{K}_{fr} M E_f \end{bmatrix} \end{aligned} \quad (27)$$

and with initial conditions $m_{nom,\beta}(k_0 - 1; k_0) = \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} \end{pmatrix} m_{nom}^{nfa}(k_0 - 1)$ and $Q_{nom}(k_0 - 1; k_0) = \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} \end{pmatrix} Q_{nom}^{nfa}(k_0 - 1) \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} \end{pmatrix}'$. At time $k_r(\beta) = k_0 + \widehat{ARL}_h(\beta)$, the reconfiguration takes place, and the mean and variance of $(\chi'_k \ \chi'_k)$, for $k \geq k_r(\beta)$ are then given by equations of the same form as (21), namely:

$$\begin{cases} m_{rec}(k+1; k_r(\beta)) = \mathcal{A}_{rec} m_{rec}(k; k_r(\beta)) \\ Q_{rec}(k+1; k_r(\beta)) = \mathcal{A}_{rec} Q_{rec}(k; k_r(\beta)) \mathcal{A}'_{rec} + \mathcal{B}_{rec} \begin{pmatrix} R_w & 0 \\ 0 & R_\nu \end{pmatrix} \mathcal{B}'_{rec} \end{cases} \quad (28)$$

with the following initial conditions $m_{rec}(k_r(\beta); k_r(\beta)) = m_{nom,\beta}(k_r(\beta); k_0)$, $Q_{rec}(k_r(\beta); k_r(\beta)) = Q_{nom,\beta}(k_r(\beta); k_0)$. The mean and variance of z in the presence of a fault for the relevant values of k , $m_{nom,\beta,z}(k; k_0)$, $Q_{nom,z}(k; k_0)$, and $m_{rec,z}(k; k_r(\beta))$, $Q_{rec,z}(k; k_r(\beta))$ are deduced in a straightforward way.

3.3 Global expression for cost \mathcal{J}

This expression is obtained by grouping the above results and accounting for the pmf of the fault occurrence time and the pdf of the fault magnitude β . One actually computes $\mathcal{J}_M = \mathcal{E} \left(\sum_{k=0}^H z_k z'_k \right)$. Then $\mathcal{J} = \text{tr}(\mathcal{J}_M)$.

$$\begin{aligned} \mathcal{J}_M &= \frac{p_{nf}}{N} \sum_{k=1}^H \left[(N_k(m_{nom,z}^{fa}(k) \star + Q_{nom,z}^{fa}(k))) \right. \\ &\quad \left. + \tilde{N}_k (m_{nom,z}^{nfa}(k) \star + Q_{nom,z}^{nfa}(k)') + \sum_{i=1}^{k-1} N_i (m_{rec,z}(k; i) \star + Q_{rec,z}(k; i)) \right] \\ &\quad + \sum_{k_0=1}^H \frac{p_{k_0}}{N} \left[\sum_{i=1}^{k_0-1} N_i \left(\sum_{k=1}^{i-1} (m_{nom,z}^{nfa}(k) \star + Q_{nom,z}^{nfa}(k)) \right. \right. \\ &\quad \left. \left. + \sum_{k=i}^H (m_{rec,z}(k; i) \star + Q_{rec,z}(k; i)) \right) \right] \end{aligned}$$

$$\begin{aligned} &+ \tilde{N}_{k_0-1} \left(\sum_{k=1}^{k_0-1} (m_{nom,z}^{nfa}(k) \star + Q_{nom,z}^{nfa}(k)) + \int p_\beta(\beta) \left(\sum_{k=k_0}^{k_r(\beta)-1} (m_{nom,\beta,z}(k; k_0) \star + Q_{nom,z}(k; k_0)) \right. \right. \\ &\quad \left. \left. + \sum_{k=k_r(\beta)}^H (m_{rec,z}(k; k_r(\beta)) \star + Q_{rec,z}(k; k_r(\beta))) \right) d\beta \right) \end{aligned} \quad (29)$$

The first three lines correspond to the contribution of the fault free trajectories. The fourth and fifth lines account for the faulty case in which a reconfiguration takes place before the fault occurrence time k_0 , due to a false alarm. Finally the remaining lines account for the faulty situation in the absence of false alarm. From a computational point of view the integral over β will have to be evaluated by Monte Carlo simulation by considering a set of values β_i , $i = 1, \dots, N_\beta$ drawn from the distribution described by $p_\beta(\beta)$.

3.4 Algorithm for computing \mathcal{J}

To solve the optimization problem (18), an appropriate numerical optimization software should be used, based on the computation of the cost function for a given value of h . Questions regarding properties of the cost, like convexity, are open at this stage. The computations of (29) can be performed using the following 6 step procedure.

1. By Monte Carlo simulation, determine $m_{nom,z}^{fa}(k)$, $Q_{nom,z}^{fa}(k)$, $m_{nom,z}^{nfa}(k)$, $Q_{nom,z}^{nfa}(k)$, $k = 1, \dots, H$ and the associated values for the state of the augmented nominal system (see (19), (20)).
2. Solve (21), with initial conditions from (19) and deduce from (22) all the terms of the third line in \mathcal{J}_M .
3. Sample N_β values from the probability density function $p_\beta(\beta)$. Solve equation (26) as well as the equations for the mean and variance of r (deduced from (13) and (15)) for fault magnitude $\beta = 1$; by linearity the values of $m_{nom,\beta}(k; k_0)$ and $m_r(k; k_0)$ for an arbitrary value of β can be obtained directly from the case $\beta = 1$.
4. For each β_i , compute $\widehat{ARL}_h(\beta_i)$ by (23), (24) on the basis of step 3.
5. Solve (28) for $k = k_r(\beta_i)$ up to H , for each β_i , and deduce $m_{rec,z}(k; k_r(\beta_i))$, $Q_{rec,z}(k; k_r(\beta_i))$
6. Compute \mathcal{J}_M by introducing the results of steps 1 to 5 into (29).

A significant simplification in the computation of the cost function can be obtained if it is assumed that the transient due to the initialization of the process equipped with its controller and supervision system has vanished, so that the state of system (6), (8), (12) is a weakly stationary random process. In this situation, $m_{nom,z}^{fa}(k)$, $Q_{nom,z}^{fa}(k)$, $m_{nom,z}^{nfa}(k)$, $Q_{nom,z}^{nfa}(k)$ are identical for all k and $N_k = N_{fa}$, a constant independent of k , while $\tilde{N}_k = N - kN_{fa}$. Besides $m_{rec,z}(k; i)$ and $Q_{rec,z}(k; i)$ only depend on the difference $(k - i)$, and similarly for $m_{nom,\beta,z}(k; i)$ and $Q_{nom,\beta,z}(k; i)$ (see Aberkane et al. [2008]).

4. NUMERICAL EXAMPLE

In this section, the proposed performance based threshold selection scheme will be illustrated using a winding machine example. The winding process is composed of a plastic web and three reels, respectively called the unwinding, pacer and rewinding reels. Each reel is coupled with a DC-motor via gear reduction. The angular speed of each reel and both tensions between the reels are measured by tachometers and tension meters. Each motor is driven by a local controller. The first control loop adjusts the motor current, while the second loop controls the angular speed. This system is linearized around the operating point $U_0 = [-0.15 \ 0.6 \ 0.15]$, $X_0 = [0.6 \ 0.5 \ 0.4]$ and we obtain the following state space representation

$$A_x = \begin{bmatrix} 0.4126 & 0 & -0.0196 \\ 0.0333 & 0.5207 & -0.0413 \\ -0.0101 & 0 & 0.2571 \end{bmatrix}, B_u = \begin{bmatrix} -1.7734 & 0.0696 & 0.0734 \\ 0.0928 & 0.4658 & 0.1051 \\ -0.0424 & -0.093 & 2.0752 \end{bmatrix}$$

$$B_w = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}; C_y = \mathbb{I}; C_z = \begin{bmatrix} \mathbb{I} \\ \mathbf{0} \end{bmatrix}; D_z = \begin{bmatrix} \mathbf{0} \\ \mathbb{I} \end{bmatrix}.$$

which corresponds to a sampled data model at sampling period $T_s = 0.1s$. The covariance matrices of the noise signals are given as follows:

$$R_w = 0.04 \times \mathbb{I}, R_\eta = 0.01 \times \mathbb{I}$$

The state vector $x_t \in \mathbb{R}^3$ is composed of the unwinding tension, the angular speed and the winding tension, respectively. The command vector is composed by the current set point of the inner loop for the unwinding reel, the voltage command of the pacer motor and the current set point the inner loop for the winding reel.

For illustration purpose, we will consider the linear quadratic performance index given by:

$$\mathcal{J} = \sum_{k=0}^{100} \mathcal{E}(z'_k z_k)$$

We assume that sensor 1 is subject to faults with a profile given as $f(k) = \beta \{ \min(\alpha(k - k_0), f_{max}) \}$, where $f_{max} = 1$, $\alpha = 0.5$ and β is a uniformly distributed random variable in $[0, 1]$. The fault time occurrence k_0 is also assumed to be uniformly distributed random variable. The residual vector R_k is obtained by stacking up $(2 + 1)$ residual samples, *i.e.* $l = 2$.

Figure 1 represents the evolution of the performance index \mathcal{J} with respect to threshold h . This figure was obtained using the algorithm of section 3.4 with $N = 10000$. It can be seen that the optimal threshold, corresponding to the minimal value of \mathcal{J} is given by $h_{opt} = \arg \min \mathcal{J} = 13$. This value is significantly lower than the threshold of a standard χ^2 test designed for a 95% or 99% false alarm probability which would respectively correspond to $h = 16.9$ or 21.7 .

5. CONCLUSION

The optimization of the threshold of a fault detection system, on the basis of the closed-loop performance of the FTC system in which it is included, is considered. A simple situation where a sensor fault is accommodated by introduction of a software sensor in place of the hardware sensor is considered. The closed loop performance is computed by resorting to a combination of Monte Carlo simulation and standard tools from stochastic system theory in order

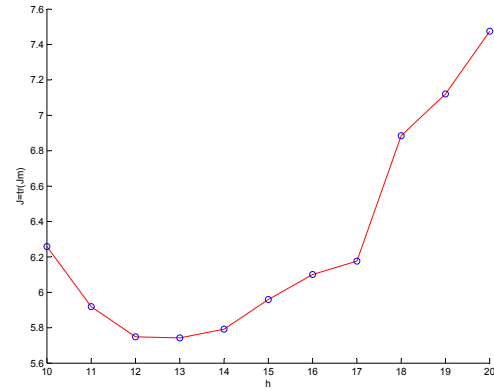


Figure 1. Performance index \mathcal{J} in function of h

to limit the computational load. The proposed algorithm is illustrated on a fault tolerant controller for a winding machine.

Acknowledgment This work was performed in the framework of a Concerted Research Action supported financially by the "Ministère de la Communauté française - Direction générale de l'Enseignement non obligatoire et de la Recherche scientifique". Support by the IUAP Programme, initiated by the Belgian State, Science Policy Office, through the Belgian Network DYSCO is also gratefully acknowledged. The scientific responsibility rests with the authors.

REFERENCES

- S. Aberkane and M. Kinnaert. Threshold selection based on closed-loop performance for fault detection. Internal report, Dpt of Control Engineering and System Analysis, ULB, 2008.
- M. Basseville and I.V. Nikiforov. Detection of Abrupt Changes: Theory and application. Prentice Hall, Englewood Cliffs, 1993.
- M. Blanke, M. Kinnaert, J. Lunze and M. Staroswiecki. Diagnosis and Fault-Tolerant Control, Second Edition. Springer, Berlin, Germany, 2006.
- A. Gani, P. Mhaskar and P.D. Christofides. Fault-tolerant control of a polyethylene reactor. Journal of Process Control, pages 439–451, 2007
- N.E. Wu, Y. Zhang and K. Zhou. Detection and accommodation of loss of control effectiveness International Journal of adaptive Control and Signal Processing, pages 775-795, 2000.
- C. Edwards and C.P. Tan. Sensor fault tolerant control using sliding-mode observers. Control Engineering Practice, pages 897–908, 2006.
- J.E. Wieringa. Statistical Process Control for Serially Correlated Data. PhD Thesis, Rijksuniversiteit Groningen, Dpt. of Econometrics, 1999.
- J. Lunze and Th. Steffen. Control reconfiguration after actuator failures using disturbance decoupling methods. IEEE Trans. automatic Control, pages 1590–1601, 2006.
- Z. Gao and P.J. Antsaklis. Stability of the pseudo-inverse method for reconfigurable control systems. Int. J. Control, pages 717–729, 1991.
- A.H. Jazwinski. Stochastic Processes and Filtering Theory. Academic Press, 1970.
- L. El Ghaoui, F. Oustry and M. AitRami. A Cone Complementary Linearization Algorithm for Static Output-Feedback and Related Problems. IEEE Transactions on Automatic Control, pages 1171-1176, 1997.