Transmission delays in residual computation

D. Llanos, M. Staroswiecki, J. Colomer and J. Meléndez

Abstract: The design of control, estimation or diagnosis algorithms most often assumes that all available process variables represent the system state at the same instant of time. However, this is never true in current network systems, because of the unknown deterministic or stochastic transmission delays introduced by the communication network. During the diagnosing stage, this will often generate false alarms. Under nominal operation, the different transmission delays associated with the variables that appear in the computation form produce discrepancies of the residuals from zero. A technique aiming at the minimisation of the resulting false alarms rate, that is based on the explicit modelling of communication delays and on their best-case estimation is proposed.

1 Introduction

Owing to the growing complexity and spatial distribution of automated systems, communication networks have become the backbone of most control architecture. As systems are required to be more scalable and flexible, they have additional sensors, actuators and controllers, often referred to as field (intelligent) devices [1, 2]. Networked control systems result from connecting these system components via a communication network such as controller area network (CAN), PROFIBUS or Ethernet.

An increasing amount of research addresses the distributed control of inter-connected dynamical processes: stability and control [3–5], decision, co-ordination and scheduling [6, 7], diagnosis of discrete event systems [8] and fault tolerance [9–11]. However, only a few studies of the impact of the communication network on the diagnosis of continuous systems have recently been published [12–14].

In model-based fault detection and isolation (FDI), a set of residuals that should be ideally zero in the fault-free case and different from zero, in the faulty case are designed [15–17]. However, in practice, residuals are different from zero, not only because of measurement noise, unknown inputs and modelling uncertainties but also because of transmission delays. Since no network can communicate instantaneously, data which are used in the residual computation do not represent the state of the system at the time of the computation. Instead, they represent the state of the system at some (often unknown) time prior to the computation. Moreover, each variable being possibly transmitted under a different transmission delay, the whole set of data that are used in the residual computation may even not be consistent with the system state at any moment prior to the computation. Therefore, residuals that should theoretically be zero in the non-faulty case might create false alarms as the result of transmission delays.

The false alarms rate can be decreased by increasing the decision threshold, at the cost of reducing the sensitivity to faults. In this paper, a technique aiming at the minimisation of the false alarms caused by transmission delays without increasing the number of missed detection is proposed. It relies on the explicit modelling of communication delays, and their most likely estimation.

The paper is organised as follows: Section 2 provides a background on fault detection and isolation based on analytical redundancy, with emphasis on the decision making logic. Section 3 presents the influence of transmission delays. An optimisation technique for the estimation of unknown delays is described in Section 4. Finally, an illustrative example is shown in Section 5.

2 Analytical redundancy for fault detection and isolation

2.1 System and faults

Consider the deterministic system modelled by

$$\dot{x}(t) = \boldsymbol{f}[x(t), u(t), \varphi(t)] \tag{1}$$

$$y(t) = \boldsymbol{g}[x(t), u(t), \varphi(t)] \tag{2}$$

where $x(t) \in R^n$, $u(t) \in R^r$, $y(t) \in R^m$ and $\varphi(t) \in R^q$ are, respectively, the state, control, output and fault vector, and f and g are given smooth vector fields.

The system normal operation on a time window $[\alpha, \beta[$ is described by

$$\varphi(t) = 0, \quad \forall t \in [\alpha, \beta[\tag{3}$$

while the occurrence of a fault at time γ is associated with

$$\exists \lambda > \gamma{:}\varphi(t) \neq 0, \quad \forall t \in [\gamma, \lambda[\tag{4}$$

2.2 Analytical redundancy relations

Analytical redundancy is based on successive derivations of the output signal (2), which together with the repeated use of (1) produces the system

$$\bar{y}(t) = G(x(t), \bar{u}(t), \bar{\varphi}(t)) \tag{5}$$

where $\bar{y}(t)$ [and also $\bar{u}(t)$ and $\bar{\varphi}(t)$] is the vector obtained by expanding $y(t)$ with its derivatives $\dot{y}(t)$, $\ddot{y}(t)$, and so on. [the

© The Institution of Engineering and Technology 2007

doi:10.1049/iet-cta:20060519

Paper first received 15th February 2006 and in revised form 18th March 2007

D. Llanos, J. Colomer and J. Meléndez are with the Institut d'Informàtica i Aplicacions, Universitat de Girona, Av. Lluis Santalo s/n E-17071 Girona, Spain

M. Staroswiecki is with SATIE UMR CNRS 8029, Ecole Normale Supérieure de Cachan, 61 avenue du Président Wilson, Cachan Cedex 94235, France

E-mail: dllanosr@eia.udg.es

highest derivation order in $\bar{y}(t)$ is not explicitly stated, because it is of no interest here].

In a second step, (5) is transformed into an equivalent system

$$\bar{y}(t) = G[x(t), \bar{u}(t), \bar{y}(t), \bar{\varphi}(t))] \iff$$

$$\begin{cases} G_1[x(t), \bar{u}(t), \bar{y}(t), \bar{\varphi}(t)] = 0 \\ G_2[\bar{u}(t), \bar{y}(t), \bar{\varphi}(t)] = 0 \end{cases} \tag{6}$$

where the equations in the subsystem G_2 are the so-called analytic redundancy relations (ARR), which are independent on the state vector $x(t)$. It can be shown that such ARR can always be found, provided the output can be derivated up to an order large enough [15–17]. The interest of these ARR is obviously that – since the state has been eliminated – they depend only on the inputs, outputs and faults, and thus providing a means to check whether the no-fault hypothesis is consistent with the observed input – outputs.

2.3 Practical determination of analytical redundancy relations

From a practical point of view, obtaining the set of equations G_2 in (6) from the original set (5) makes use of a projection operator when system (1), (2) is linear (this is the parity space technique, see for example [15, 17]). However, for more general cases, it rests on elimination theory (see for example [18] for the case where (1), (2) is polynomial).

It is also worth to notice that ARR are not uniquely defined. Indeed, any linear or nonlinear combination of analytical redundancy relations is also an analytical redundancy relation.

2.4 Fault detection and isolation

2.4.1 Computation and evaluation forms: Let the subsystem G_2 be decomposed as

$$G_2[\bar{u}(t), \bar{y}(t), \bar{\varphi}(t)] = G_c[\bar{u}(t), \bar{y}(t)] - G_e[\bar{u}(t), \bar{y}(t), \bar{\varphi}(t)]$$

where for all input/output pairs $u(t)$, $y(t)$ are associated with systems (1) and (2)

$$G_e[\bar{u}(t), \bar{y}(t), 0] = 0 \tag{7}$$

Then condition $G_2 = 0$ can be written as

$$r(t) \triangleq G_c[\bar{u}(t), \bar{y}(t)] \tag{8}$$

$$r(t) = G_e[\bar{u}(t), \bar{y}(t), \bar{\varphi}(t)] \tag{9}$$

where $r(t)$ is the residual vector, and (8) and (9) are, respectively, its computation and evaluation form. The first one describes how the residual value is obtained from the system inputs and outputs. The latter describes how the resulting value depends on faults.

According to (8) and (9), the fault detection and isolation procedure is decomposed into two steps. The first one is residual computation where the residual value is computed from the known variables, using the computation form (8). The second step is residual evaluation, that includes fault detection and fault isolation.

2.4.2 Fault detection: Given a time window $[\alpha, \beta[$, the fault detection problem is defined as follows: given the residual $r(t), t \in [\alpha, \beta[$, select the most likely hypothesis between H^0_{system} and H^1_{system} where

$$H^0_{\text{system}} \quad : \quad \varphi(t) = 0, \forall t \in [\alpha, \beta[$$

$$H^1_{\text{system}} \quad : \quad \exists[\gamma, \lambda[\subseteq [\alpha, \beta[: \varphi(t) \neq 0, \quad \forall t \in [\gamma, \lambda[$$

Using (7) and (9), the simplest implementation of a fault detection procedure is obtained by checking the residual value against zero at each time t (by a slight abuse of notation, the time intervals $[\alpha, \beta[$ and $[\gamma, \lambda[$ are no longer mentioned).

$$[H^0_{\text{system}} \Longrightarrow r(t) = 0] \iff [r(t) \neq 0 \Longrightarrow H^1_{\text{system}}] \tag{10}$$

For the sake of simplicity, only perfect deterministic models have been considered so far that result in (10) being indeed true. However, measurement noise, unknown inputs and model uncertainties will result in residuals being never zero even in normal operation. This can be taken into account in a more realistic procedure which extends (10) as follows

$$\left[H^0_{\text{system}} \Longrightarrow r(t) \in \mathcal{N}(0)\right] \iff \left[r(t) \notin \mathcal{N}(0) \Longrightarrow H^1_{\text{system}}\right] \tag{11}$$

where $\mathcal{N}(0)$ is some neighbourhood of zero. Note that false alarms – $r(t) \notin \mathcal{N}(0)$ under H^0_{system} – and missed detections – $r(t) \in \mathcal{N}(0)$ under H^1_{system} – are possible. The design of a set $\mathcal{N}(0)$ that guarantees both a low false alarm and a low missed detection-rate is the central problem of statistical decision making [19–21].

2.4.3 Fault isolation: Fault isolation rests on special properties of the residual evaluation form (directional residual and structured residuals) are not developed here (see for example [15, 16, 18] for good presentations): it is assumed in the sequel that the residual vector $r(t)$ has satisfactory detection/isolation properties.

3 Influence of transmission delays

3.1 Data decomposition in distributed systems

In distributed control systems, the residual computation form is implemented as an algorithm in one node of the network. At each time t, its input data are noted as

$$z(t) \triangleq \begin{pmatrix} \bar{u}(t) \\ \bar{y}(t) \end{pmatrix}$$

According to the overall system distributed architecture, $z(t)$ is decomposed into a set of subvectors $z_i(t)$, $i \in I$ and the computation form of the residual vector writes

$$r(t) = G_c\big(z_i(t), i \in I\big) \tag{12}$$

The subvectors $z_i(t)$ are such that all variables in $z_i(t)$ are transmitted in one single packet through the communication network. Note that this does not imply that all the variables produced at a given node are transmitted in one single packet. As shown in Fig. 1

$$z(t) = [z_1^{\mathrm{T}}(t), z_2^{\mathrm{T}}(t), z_3^{\mathrm{T}}(t)]^{\mathrm{T}}$$

where z_1 is produced and transmitted by node 1 (a smart sensor), while $z_2 \cup z_3$ are produced by node 2 (a local controller). The data are decomposed into packets z_2 and z_3 for their transmission through the communication system.

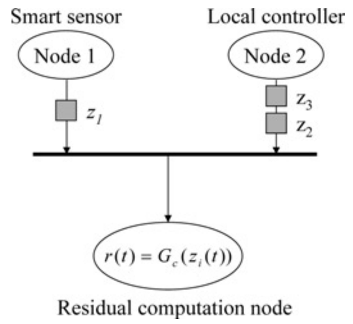

Fig. 1 *Data decomposition in distributed system*

3.2 Incidence of transmission delays

Owing to the transmission delays, the data $z_i(t)$ generated at the production nodes and the data $\hat{z}_i(t)$ available at the residual computation node must be distinguished. One obviously has

$$\hat{z}_i(t) = z_i(t - \delta_i), \quad i \in I \tag{13}$$

where $\delta_i \in R^+$ is the transmission delay, that is the data z_i was produced at time $t - \delta_i$, and it was received only at time t. Transmission delays δ_i may be time dependent and generally unknown. The normal operation of the communication network on a given time window $[\alpha, \beta[$ can be described by a very simple deterministic model. Namely, the maximum delay Δ is assumed to be known

$$\begin{aligned}&\left[H^0_{\text{network}} \Longrightarrow \forall i \in I, \forall t \in [\alpha, \beta[: \delta_i \leq \Delta\right] \\ &\quad \Longleftrightarrow \left[\exists i \in I, \exists [\gamma, \lambda[\subseteq [\alpha, \beta[: \delta_i > \Delta \Longrightarrow H^1_{\text{network}}\right]\end{aligned} \tag{14}$$

If communication delays are not taken into account, residual computation can be performed as

$$r(t) = G_c\left[\hat{z}_i(t), i \in I\right] \tag{15}$$

but using data which are taken from the system at different time instants would obviously result in false alarms.

Taking into account the communication delays by using future values of the arguments, as in (16), is obviously impossible.

$$r(t) = G_c\left[\hat{z}_i(t + \delta_i), i \in I\right] \tag{16}$$

Finally, the only possibility to obtain a feasible algorithm is to 'synchronise' the data by using a delay τ as follows

$$\rho(t) = G_c\left[\hat{z}_i(t - \tau + \delta_i), i \in I\right] \tag{17}$$

where $\rho(t)$ is the residual available at the residual computation node. Note that at a time t one in fact computes the value that the residual had at time $t - \tau$

$$\rho(t) = r(t - \tau) \tag{18}$$

The delay τ must obviously satisfy

$$\tau \geq \max_{i \in I} \delta_i \tag{19}$$

4 Decision procedure under unknown transmission delays

When the vector of transmission delays

$$\delta = (\delta_i, i \in I)$$

is perfectly known, the decision procedure associated with (11) can be directly run by choosing

$$\tau = \|\delta\|_\infty$$

from (18) it follows that

$$\left[H^0_{\text{system}} \Longrightarrow \rho(t) \in \mathcal{N}(0)\right] \Longleftrightarrow \left[\rho(t) \notin \mathcal{N}(0)\right] \Longrightarrow H^1_{\text{system}}\Big] \tag{20}$$

however when $\rho(t) \notin \mathcal{N}(0)$ the fault detection process is delayed by τ.

When transmission delays are unknown, the decision has to be taken in the presence of the so-called nuisance parameters δ. From (14) and (20), the following decision logic is true

$$\begin{aligned}&[H^0_{\text{system}} \wedge H^0_{\text{network}}] \Longrightarrow \exists \delta:[(\|\delta\|_\infty \leq \Delta) \wedge (\rho(t) \in \mathcal{N}(0))] \\ &\quad \Longleftrightarrow \forall \delta:[(\|\delta\|_\infty > \Delta) \vee (\rho(t) \notin \mathcal{N}(0)] \\ &\quad \Longrightarrow [H^1_{\text{system}} \vee H^1_{\text{network}}]\end{aligned} \tag{21}$$

This decision logic expresses that the non-existence of a vector of transmission delays, δ such that (1) $\|\delta\|_\infty \leq \Delta$ and (2) the residual lies inside $\mathcal{N}(0)$ evidences that the system, the network, or both do not operate properly. Even though both faults in the network and in the system can be detected, they cannot be isolated from each other in the absence of extra information.

4.1 Estimating the transmission delays

Let the system fault detection neighbourhood $\mathcal{N}(0)$ be defined by

$$\mathcal{N}(0) = \left\{\rho : J(\rho) \leq \sigma\right\}$$

where $\forall \rho \neq 0, J(\rho) > 0, J(0) = 0$, for example $J(\rho) = \rho^{\mathrm{T}} Q \rho$ with $Q > 0$ and $\sigma > 0$ is a given decision threshold. Checking the property $H^0_{\text{system}} \wedge H^0_{\text{network}}$ can be done by solving the following optimisation problem

$$\breve{\delta} = \arg \min_{\|\delta\|_\infty \leq \Delta} J[\rho(t)] \tag{22}$$

where $\rho(t) = G_c[\hat{z}(t - \tau + \delta)]$. Let $\breve{\tau} = \|\breve{\delta}\|_\infty$, $\breve{\rho}(t) = G_c[\hat{z}(t - \breve{\tau} + \breve{\delta})]$ and $\breve{J}(t) = J[\breve{\rho}(t)]$, then the following interpretation holds

(i) $\breve{\delta}$ is the 'most likely' vector of admissible delays in the sense that the function $\breve{J}(t)$ associated with the delayed residual $\breve{\rho}(t)$ is minimum
(ii) This residual may or may not be compatible with the hypothesis that the system operates in a nominal way, and therefore the decision logic (21) becomes

$$\breve{J}(t) > \sigma \Longrightarrow H^1_{\text{system}}$$

Finally, it should be noted that the estimation of $\breve{\delta}$ by (22) implements a sufficient condition-based decision logic. Indeed, if the minimal value $\breve{J}(t)$ associated with $\breve{\delta}$ does not satisfy $\breve{J}(t) \leq \sigma$, then no other estimation will do. However, the set $\{\delta:[\|\delta\|_\infty \leq \Delta] \wedge [\rho(t) \in \mathcal{N}(0)]\}$ may contain more than one element.

4.2 Searching for a minimum

The cost function $J(\rho(t)) = J\{G_c[\hat{z}(t-\tau+\delta)]\}$ is in general a nonlinear function of the adjustable parameters δ, and its minimum can be found using the well-known iterative search methods [24]. However, since the problem is to be solved in real time, it is of interest to study the conditions under which the estimation $\check{\delta}$ can be found quickly and accurately. Note that even when a dynamic feedback control loop is involved, the on-line search for a minimum, being a part of the FDI algorithm, can be run at a much lower frequency than the one associated with the control computation, for systems where faults are not critical (should faults be critical, it can be assumed that the network would have been designed so as to make transmission delays negligible).

4.2.1 Persistent excitation condition: Assuming that all functions involved are differentiable, the solution of the optimisation problem (22) satisfies the necessary condition

$$\frac{\partial J[\rho(t)]}{\partial \delta_i} + \mu_i = 0, \quad i \in I \tag{23}$$

where μ_i is the Khün and Tücker parameter associated with the inequality constraint $\check{\delta}_i \leq \tau$. This system can be solved for $\check{\delta}$ if its Jacobian is not too ill conditioned in a neighbourhood of the optimum. From

$$\frac{\partial J[\rho(t)]}{\partial \delta_i} = \left[\frac{\mathrm{d}\hat{z}_i}{\mathrm{d}\delta_i}(t-\tau+\delta_i)\right]^{\mathrm{T}}\left[\frac{\mathrm{d}\rho(t)}{\mathrm{d}\hat{z}_i}\right]\frac{\mathrm{d}J[\rho(t)]}{\mathrm{d}\rho}$$

it is seen that there are some system trajectories that produce a rank defective Jacobian, namely when $z_i(t)$ is constant for some $i \in I$. Therefore, for delay estimation, it is necessary that a persistent excitation condition be satisfied, such that no transmitted packet of variables is constant over time.

When the persistent excitation condition is not satisfied, the previous estimated value of the delay can be used to compute $\rho(t)$.

4.2.2 Local minima: The search for a minimum is started out from an initial guess on the parameters $\delta^{(0)}$ and in general it will converge towards a local minimum. Starting with zero at the very beginning and taking the last estimate of the transmissions delays as the initial guess for the next estimation seems to be a good approach when the exploitation conditions of the network do not change at a faster rate than the rate at which residuals are computed. Converging towards a local minimum is a source of false alarms, since the estimation $\check{\delta}$ may lead to $\check{\rho}(t) \notin \mathcal{N}(0)$ while the global minimum, say δ^*, would have provided $\rho^*(t) \in \mathcal{N}(0)$. In special cases, such as linear systems and convex cost functions, global minima will be found, but in more general cases, algorithms that avoid getting trapped in a local minimum are to be used [22].

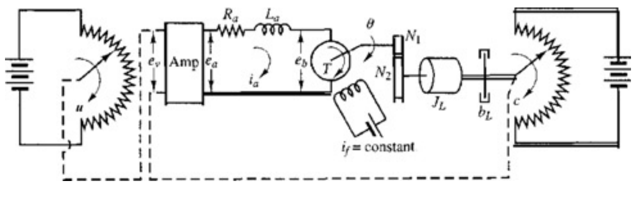

Fig. 2 *Schematic diagram of a control position for a DC motor*

5 Illustrative example

Fig. 2 depicts the schematic diagram of a control position for a DC motor. The aim of this system is to control the position $c(t)$ of the mechanical load according to the reference position $u(t)$. This system has been taken from [23].

The transfer function in open loop is

$$\frac{C(s)}{E_v(s)} = \frac{K_m}{s(T_m s + 1)} \tag{24}$$

where $C(s) = \mathcal{L}[c(t)]$ and $E_v(s) = \mathcal{L}[e_v(t)]$, $e_v(t)$ is the error between the output and the input positions. K_m and T_m are known parameters of the system.

Let

$$r(t) = T_m \frac{\mathrm{d}^2 c(t)}{\mathrm{d}t^2} + \frac{\mathrm{d}c(t)}{\mathrm{d}t} - K_m e_v(t) \tag{25}$$

be the residual computation form associated with the system, where $c(t)$ and $e_v(t)$ are assumed to be produced in two different nodes of a distributed system (Fig. 3).

The data available at the residual computation node are $\hat{e}_v(t)$ and $\hat{c}(t)$, and the actually computed residual is

$$\rho(t) = T_m \frac{\mathrm{d}^2 c(t-\tau+\delta_1)}{\mathrm{d}t^2} + \frac{\mathrm{d}c(t-\tau+\delta_1)}{\mathrm{d}t} - K_m e_v(t - \tau + \delta_2) \tag{26}$$

where δ_i are the delays through the network and τ is their maximum value (19).

Simulations have been performed with $K_m = 5.5\ \mathrm{s}^{-1}$ and $T_m = 0.13$ s. $u(t)$ has been modelled as an unit step signal and $c(t)$ as its response. δ_1 and δ_2 are uniformly distributed in the interval [0, 0.1] s.

Under $H^0_{\text{system}} \wedge H^0_{\text{network}}$, one obviously obtains $\rho(t) = 0$ when δ_i are replaced by their actual values. When these values are unknown, the admissible delays $\check{\delta}_i$ are estimated, by solving the optimisation problem

$$\min \rho^2(t), \text{ under the constraint } \quad \delta_i \in [0, 0.1]\,\mathrm{s} \tag{27}$$

Matlab has been used for simulation and optimisation. The function fmincon has been used for delays estimation. This function uses a sequential quadratic programming method [24], and is suitable for finding a minimum of a constrained nonlinear multivariable function. Initial condition, δ^0, for the algorithm has been fixed at zero.

In order to study the benefits of delay estimation to optimise residual computation, neither unknown inputs (noise or disturbances) nor uncertainties have been considered. The analysis has been focused on the performance of the method to reduce false alarms in the absence of system faults and to reduce the missed detection rates in presence of a sensor fault.

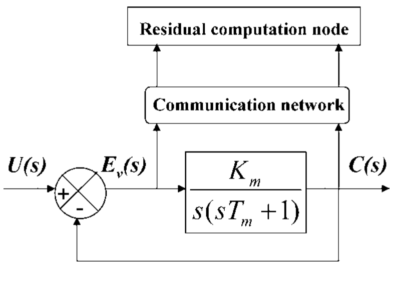

Fig. 3 *Block diagram of the system*

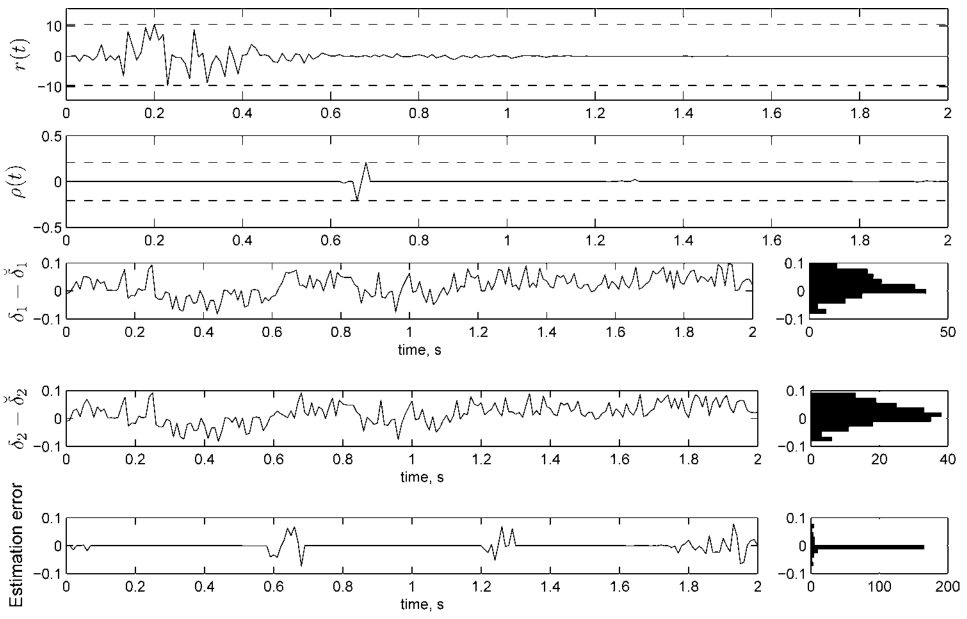

Fig. 4 *Residual in fault-free situation, comparisons of the actual and estimated delays and the estimation error*

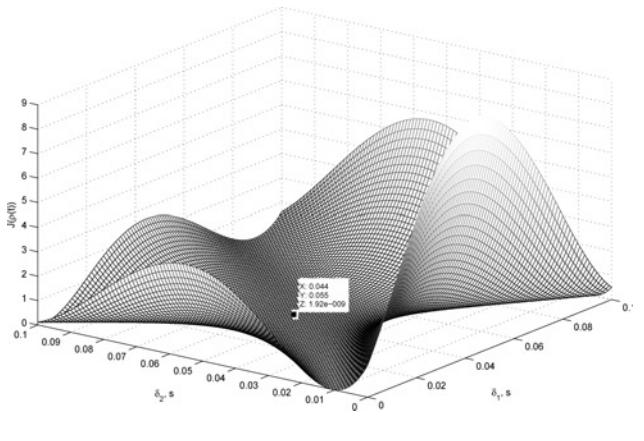

Fig. 5 *Cost function J[ρ(t)]*

5.1 Fault-free situation

Fig. 4 depicts the residual computed without and with the delay estimation in a fault-free situation [$r(t)$ above and $\rho(t)$ below]. Notice that during the transient response, defined by the interval [0, 1.03] s, the residuals are more sensitive to the actual delays than during the steady-state time window. It is well known that false alarms can always be avoided (e.g. by never firing any alarm). This can be seen in Fig. 4, which shows that false alarms are avoided in both cases, but at the cost of four orders of magnitude on the decision threshold (± 10 instead of ± 0.25). This means worse performances when faults will be present (larger missed detection rate, bigger size for faults to be detectable).

The comparisons of actual (δ_1 and δ_2) and estimated ($\breve{\delta}_1$ and $\breve{\delta}_2$) delays, and their distributions, are depicted in Fig. 4. Owing to the lack of a time reference, it seems that estimations are not well achieved. In order to introduce a time reference, the estimation error is computed as the difference between the comparisons of the actual and estimated delays (Fig. 4). The

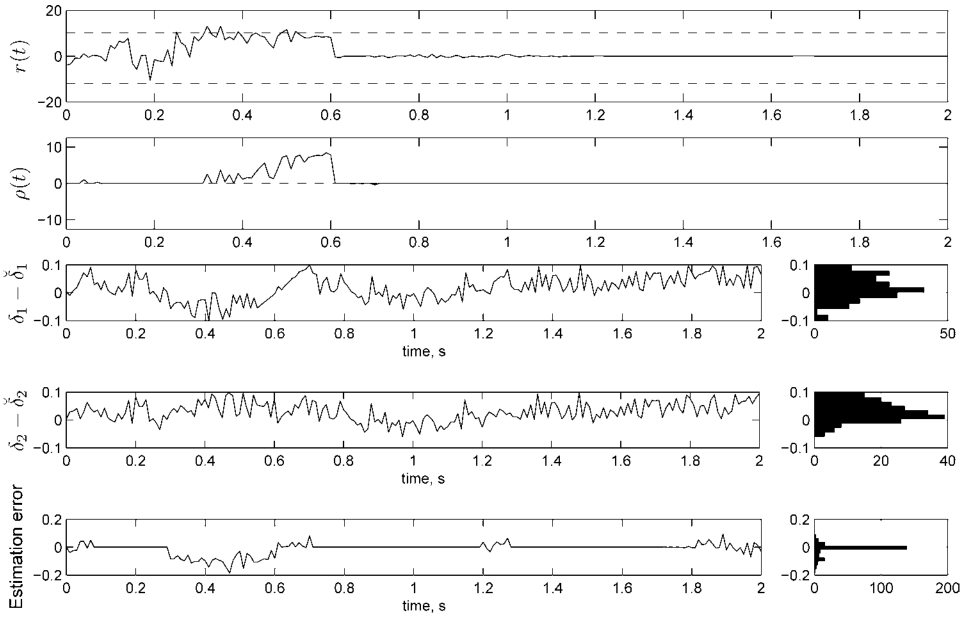

Fig. 6 *Residual in faulty situation, comparisons of the actual and estimated delays and the estimation error*

estimation error is close to zero, and it increases during the steady-state time window because the excitation condition is not fulfilled, nevertheless it does not affect the computation of the residual $\rho(t)$ as can be seen in Fig. 4.

Fig. 5 depicts the quadratic cost function $J[\rho(t)]$ to be minimised at one time instant, in that case the actual delays were $\delta_1 = 0.044$ s and $\delta_2 = 0.055$ s, the dot on the figure is the minimum of the cost function $J[\rho(t)] = 1.92 \times 10^{-9}$.

5.2 Faulty system situation

Fig. 6 represents residuals $r(t)$ and $\rho(t)$ in the presence of an additive fault introduced on the sensor $e(t)$ as a constant 0.2 bias during the time window [0.3, 0.6] s. The effect of this fault on residual $r(t)$ is not large enough to overpass the thresholds, causing missed detections during almost all the fault instants. On the other hand, the lower figure depicts how $\rho(t)$ allows a proper detection. The estimation error increases significantly during the fault instants, but it does not affect the detection of the fault.

6 Conclusion

In model-based FDI a set of residuals is computed, which should be ideally zero in the fault-free case and different from zero, in the faulty case. However, in practice, residuals are different from zero, not only because of faults but also because of measurement noise, unknown inputs modelling uncertainties and transmission delays in distributed systems. The use of these residuals will produce false alarms and missed detections.

In this paper, it is shown that when transmission delays are known, it is possible to take them into account in the residual computation, and thus introducing a delayed but otherwise unchanged decision and avoiding false alarms as a result of delays. However, when delays are unknown, it is necessary to estimate them in order to compensate for their effect in the decision procedure. In this case, based on a very rough model of the delays, the paper proposes to address the problem as an optimisation problem. A search algorithm is used for the delay estimation by minimising the residual under the constraints given by the transmission model. When the persistent excitation condition is fulfilled, the delays estimation can be carried out giving reliable results and avoiding false alarms.

The efficiency of the proposed approach is illustrated by a simple example where a set of different dynamics and fault magnitudes have been simulated. Current research addresses more complex transmission models, and more complex sources of errors (measurement noise and transmission errors).

7 References

1 Lee, D., Allan, J., Thompson, H.A., and Bennett, S.: 'PID control for a distributed system with a smart actuator', *Control Eng. Practice*, 2001, **9**, (11), pp. 1235–1244

2 Staroswiecki, M.: 'Intelligent sensors: a functional view', *IEEE Trans. Ind. Inform.*, 2005, **1**, (4), pp. 238– 249

3 Dong, Y., Qing-Long, H., and Chen, P.: 'State feedback controller design of networked control systems', *IEEE Trans. Circuits Syst. II. Analog Digit. Signal Process.*, 2004, **51**, (11), pp. 640–644

4 Montestruque, L.A., and Antsaklis, P.J.: 'On the model-based control of networked systems', *Automatica*, 2003, **39**, pp. 1837–1843

5 Montestruque, L.A., and Antsaklis, P.J.: 'Stability of model-based networked control systems withtime-varying transmission times', *IEEE Trans. Autom. Control*, 2004, **49**, (9), pp. 1562–1572

6 Tipsuwan, Y., and Chow, M.Y.: 'Control methodologies in networked control systems', *Control Eng. Pract.*, 2003, **11**, (10), pp. 1099–1111

7 Walsh, G.C., and Ye, H.: 'Scheduling of networked control systems', *IEEE Control Syst. Mag.*, 2001, **21**, (1), pp. 57–65

8 Neidiga, J., and Lunze, J.: 'Decentralised diagnosis of automata networks'. Proc. 16th IFAC World Congress, Prague, July 2005

9 Jalote, P.: 'Fault tolerance in distributed systems' (Prentice- Hall, Upper Saddle River, NJ, 1994)

10 El-Farra, N.H., Gani, A., and Christofides, P.D.: 'Fault-tolerant control of process systems using communication networks', *AIChE J.*, 2005, **51**, (6), pp. 1665–1682

11 Patankar, R.P.: 'A model for fault-tolerant networked control system using TTP/C communication', *IEEE Trans. Veh. Technol.*, 2004, **53**, (5), pp. 1461–1467

12 Zhang, P., Ding, S.X., Frank, P.M., and Sader, M.: 'Fault detection of networked control systems with missing measurements'. Proc. 5th Asian Control Conf., Melbourne, 2004

13 Ding, S.X., and Zhang, P.: 'Observer-based monitoring of distributed networked control systems'. Proc. 1st NeCST Workshop Networked Control Systems and Fault Tolerant Control, Ajaccio, France, 2005

14 Fang, H., Ye, H., and Zhong, M.: 'Fault diagnosis of networked control systems'. Proc. 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Beijing, P. R. China, 2006

15 Chen, J., and Patton, R.J.: 'Robust model-based fault diagnosis for dynamic systems' (Kluwer, Boston, MA, 1999)

16 Gertler, J.: 'Fault detection and diagnosis in engineering systems' (Marcel Dekker, New York, 1998)

17 Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M.: 'Diagnosis and fault-tolerant control' (Springer Verlag, Berlin, 2003)

18 Staroswiecki, M., and Comtet-Varga, G.: 'Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems', *Automatica*, 2001, **37**, (5), pp. 687–699

19 Basseville, M., and Nikiforov, I.: 'Detection of abrupt changes: theory and applications' (Prentice-Hall, Englewood Cliffs, NJ, 1993)

20 Brodsky, B.E., and Darkhovsky, B.S.: 'Non-parametric statistical diagnosis: problems and methods' (Kluwer, Dordrecht, 2000)

21 Nikiforov, I., Staroswiecki, M., and Vozel, B.: 'Duality of analytical redundancy and statistical approach in fault diagnosis'. Proc. 13th IFAC World Congress, San Francisco, 1996

22 Goldberg, D.E.: 'Genetic algorithms in search, optimization, and machine learning' (Addison-Wesley, Boston, MA)

23 Ogata, K.: 'Modern control engineering' (Prentice-Hall, Englewood Cliffs, NJ, 1997)

24 Reklaitis, G.V., Ravindran, A., and Ragsdell, K.M.: 'Engineering optimization: methods and applications' (Wiley, New York, 1983)