Model-Based Robust Fault Detection Using a Forward – Backward Test

Alexandru Stancu, Vicenç Puig, Joseba Quevedo

Automatic Control Department - Campus de Terrassa
Universidad Politécnica de Cataluña (UPC)
Rambla Sant Nebridi, 10. 08222 Terrassa (Spain)
{Alexandru.Stancu, Vicenc.Puig, Joseba.Quevedo}@upc.es

Abstract. The problem of robust fault detection using interval observers has been mainly addressed checking if the measured behaviour is inside the region of possible states. This task can be computationally expensive because the interval observers can be affected by the wrapping effect. In this paper, a mixed approach consisting in computing a computationally cheaper inner approximation of the state region, based only on simulating vertices of parameter uncertainty region (forward test), is combined with a backward consistency check when the real measured behaviour falls outside this inner solution (backward check). The backward check is implemented using interval constraint satisfaction propagation algorithms which can perform efficiently in deciding if the measured state is consistent with the interval model. The classical alternative to this backward check will force to solve a global optimization problem, or equivalently, a global consistency problem. Finally, this approach will be tested on the DAMADICS FDI benchmark.

1 Introduction

Model-based fault detection is based on the use of mathematical models of the monitored system. However, modelling errors and disturbances in complex engineering systems are inevitable. Therefore, there is a need to develop robust fault detection algorithms. The ***robustness*** of a fault detection system means that it must be only sensitive to faults, even in the presence of model-reality differences (Chen and Patton,1999). One of the approaches to robustness, known as ***active***, is based on generating residuals which are insensitive to uncertainty, while at the same time sensitive to faults. This approach has been extensively developed these last years by several researchers using different techniques: unknown input observers, robust parity equations, H_∞, etc. (Chen and Patton, 1999). But, according to Gertler (1998), in case of models with uncertainty located in the parameters, known as ***interval models***, ***perfect decoupling*** of the residuals from uncertainties is only possible in a limited number of model parameters. In case of unlimited number of uncertain parameters, there is a second approach, called ***passive***, that enhances the robustness of the fault detection system at the decision-making stage, mainly propagating the effect of the parameter uncertainty to the residual that can be used as an adaptive threshold. Actually, several research groups are following this approach, also known, as the ***bounding-approach***, because of the use of bounds to describe the parameter and residual uncertainty (Puig, 2002). In general, computing an exact threshold is time consuming due to the optimisation problem that must be solved at each time instant. Moreover, considering the problems presented in Puig (2003b) when dealing with interval observers (the wrapping effect, the interval function range evaluation, the uncertain parameter time dependency), a new algorithm to detect faults based on forward/backward test is presented in *Section 3*. Basically, this algorithm consists in two steps: first a forward test based on checking if measurements belong to the inner solution of the observation set and a backward test based on a consistency test between measurements and the interval model. Forward test cannot assure that a fault occurred when measurements are outside the inner solution computed by the vertex simulation. To check whether or not this measure signals a fault, a consistency test must be performed to verify if there are system parameters that can explain this output value. This stage represents the backward test and is equivalent with a system identification for one data.

Finally, the algorithm will be tested in order to detect faults on the DAMADICS benchmark problem.

2. Problem formulation

2.1 Residual generation and robustness issues

A ***residual generator*** can be constructed by

$$\boldsymbol{r}(k)=\boldsymbol{y}(k)-\hat{\boldsymbol{y}}(k) \tag{1}$$

where: $\boldsymbol{r}(k)$ is the vector of residuals, $\boldsymbol{y}(k)$ and $\hat{\boldsymbol{y}}(k)$ are vectors of real and estimated measures. Ideally, the residuals should only be affected by the faults. However, the presence of disturbances, noise and modeling errors causes the residuals to become nonzero and thus interferes with the detection of faults. Therefore, the fault detection procedure must be ***robust*** in the face of these undesired effects. Robustness can be achieved in the residual generation (***active robustness***) or in the decision making stage (***passive robustness***) (Chen and Patton, 1999). The passive approach is based not in avoiding the effect of uncertainty in the residual, but in propagating the effect of uncertainty to the residual. If the residual

$$\boldsymbol{r}(k)=\boldsymbol{y}(k)-\hat{\boldsymbol{y}}(k)\in\left[\underline{\boldsymbol{r}}(k),\overline{\boldsymbol{r}}(k)\right] \tag{2}$$

no fault can be indicated, because the residual value can be due to the parameter uncertainty.

2.2 Passive Robustness Based on Interval Observers

Instead of using directly the model of the monitored system, an observer for this system will be considered. The ***non-linear interval observer*** equation without noise, faults and disturbances is:

$$\begin{aligned}\hat{\boldsymbol{x}}(k+1) &= \boldsymbol{g}(\hat{\boldsymbol{x}}(k),\boldsymbol{u}(k),\boldsymbol{\theta}) + \boldsymbol{K}(\boldsymbol{y}(k)-\hat{\boldsymbol{y}}(k)) \\ \hat{\boldsymbol{y}}(k) &= \boldsymbol{h}(\hat{\boldsymbol{x}}(k),\boldsymbol{u}(k),\boldsymbol{\theta})\end{aligned} \tag{3}$$

The evaluation of the interval for estimated measurements provided by the interval observer (3): $\left[\underline{\boldsymbol{y}}(k),\overline{\boldsymbol{y}}(k)\right]$ in order to evaluate the interval for residuals: $\left[\underline{\boldsymbol{r}}(k),\overline{\boldsymbol{r}}(k)\right]$ will be computed by means of a ***worst-case observation***. It consists in computing a region of confidence for system state set $\hat{\boldsymbol{X}}_{k+1}$, based on the confidence region for the system parameters $\boldsymbol{\Theta}$, the previous confidence region for the system state set $\hat{\boldsymbol{X}}_k$ (in the case of one step algorithms), or the previous confidence regions for the system state set $\hat{\boldsymbol{X}}_k,\ldots,\hat{\boldsymbol{X}}_{k-L}$ (in the case of sliding time window) and the measures available.

The observer equation (3) can be reorganised as a system with one output and two inputs, according to (4). Then, worst-case observation can be formulated as a ***worst-case simulation***

$$\begin{aligned}\hat{\boldsymbol{x}}(k+1) &= \boldsymbol{g_o}(\hat{\boldsymbol{x}}(k),\boldsymbol{u}_o(k),\boldsymbol{\theta}) \\ \hat{\boldsymbol{y}}(k) &= \boldsymbol{h}(\hat{\boldsymbol{x}}(k))\end{aligned} \tag{4}$$

where: $\boldsymbol{u}_o(k) = [\boldsymbol{u}(k) \quad \boldsymbol{y}(k)]^t$ and
$\boldsymbol{g_0}\left(\hat{\boldsymbol{x}}(k),\boldsymbol{u}_0(k),\boldsymbol{\theta}\right) = \boldsymbol{g}\left(\hat{\boldsymbol{x}}(k),\boldsymbol{u}(k),\boldsymbol{\theta}\right) + K\boldsymbol{y}(k) - K\boldsymbol{h}\left(\hat{\boldsymbol{x}}(k),\boldsymbol{u}(k),\boldsymbol{\theta}\right)$ is the observer non-linear function.

3. Forward-Backward algorithm

Considering the problems and computational complexity that appears in interval simulation/observation presented in Puig (2003b), when determining the interval for estimated measurements, a new algorithm for fault detection is proposed. The aim of this algorithm is not in computing the exact interval for estimated measurements but instead on verifying if they are consistent with real measurements.

This algorithm is based on a two decision tests. First test checks if real measurements are inside to inner approximation of the interval for estimated measurements guaranteeing that free faulty situations are detected. Second test is activated when measurements are outside the inner approximation of the interval for estimated measurements. In this case, the measurement is used to invalidate the interval model detecting the fault in case of invalidation is confirmed. This test guarantees that any fault that invalidates the interval model is detected.

3.1 Forward test based on Vertex Simulation (*Kolev's algorithm, 1993*)

The forward test requires an inner solution of the interval for estimated measurements. The vertex simulation (Kolev's algorithm) will produce an inner solution, i.e. a subset of solutions when the interval system is non-monotonic respect all the states and the exact solution in case of the monotonic systems (systems with isotony property (Cugueró, 2002)).

Kolev's algorithm provides the ***inner solution*** for the interval observation problem solving it approximately by determining the interval vector $\mathbf{Y}(k)=\left[\underline{\boldsymbol{y}}(k),\overline{\boldsymbol{y}}(k)\right]$ for the time interval [0,k]. This interval can be determined by solving the following global optimisation problems:

$$\begin{aligned}\overline{\boldsymbol{y}}(k) &= \max\ \boldsymbol{f}(k;\boldsymbol{A}(\boldsymbol{\theta}),\boldsymbol{B}(\boldsymbol{\theta})) \quad \textit{and} \\ \underline{\boldsymbol{y}}(k) &= \min\ \boldsymbol{f}(k;\boldsymbol{A}(\boldsymbol{\theta}),\boldsymbol{B}(\boldsymbol{\theta}))\end{aligned} \tag{5}$$

subject to: $\theta \in V(\boldsymbol{\Theta})$

where: $V(\boldsymbol{\Theta})$ denotes all vertices of the parameter uncertain *p*-dimensional vector. The interval vector ***Y(k)*** provides an inner solution due to the fact that

$$\underline{\boldsymbol{y}}_i(k) \ge \underline{\boldsymbol{x}}_i(k) \tag{6}$$

$$\overline{\boldsymbol{y}}_i(k) \le \overline{\boldsymbol{x}}_i(k) \tag{7}$$

for the time k, i = 1,..,n and $\square \boldsymbol{X}(k)=\left[\underline{\boldsymbol{x}}(k),\overline{\boldsymbol{x}}(k)\right]$. And moreover, $\boldsymbol{Y}$(k) = □ V($\boldsymbol{X}$(k))

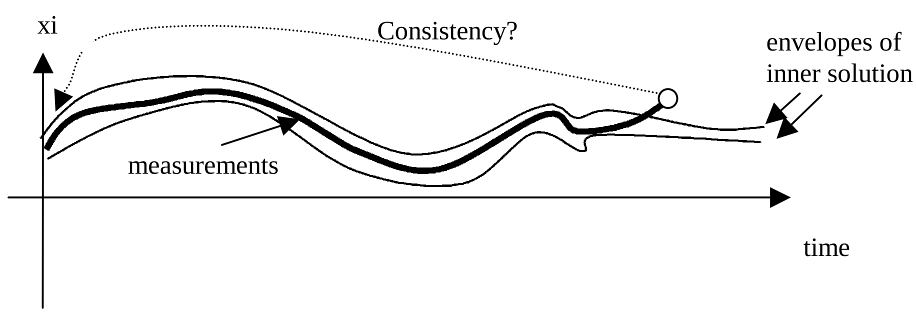
where: $\boldsymbol{X}(k)=\left\{\boldsymbol{x}(k)=\boldsymbol{f}(k;\boldsymbol{A}(\boldsymbol{\theta}),\boldsymbol{B}(\boldsymbol{\theta})):\boldsymbol{\theta}\in V(\boldsymbol{\Theta})\right\}$

The inner solution coincides with the interval hull of the solution set for some particular systems, for example, in the case of systems without the wrapping effect, according to Nickel (1985). And, moreover, according to Kolev (1993), for a constant input $u(k)=u$, the inner solution coincides over the time intervals $[0,k_1]$ and $[k_2,\infty)$ with the exact solution.

3.2 Backward Test based on Consistency Test (Identification) (Jaulin, 2001)

The consistency test is based on solving a constraint satisfaction problem (CSP) over interval domains. Consistency techniques are very efficient to contract the domain for the variables involved. Also, consistency techniques can be used with success in real time for checking the consistency of the measured data with the system parameters.

The backward test is equivalent to check the measurements consistency with the interval model. If a measurement coming from the sensor is outside the inner solution the consistency test will be perform in order to check if this measurement is consistent with the interval model. If is not consistent, then we can assure that a fault was occurred. If there are consistency between the measurement and the interval model then we cannot decide anything due to the local consistency was performed. In this case a global consistency is needed (time consuming), or this consistency test must be combined with another strategy in order to decide if the measurement represent a fault or not. Intuitively, to illustrate the idea of this algorithm in the figure below is shown that when a measurement falls outside the inner solution the consistency test will be performed in order to validate or not the interval model.



For the system (4) the consistency problem can be represented by a constraint satisfaction problem, where the set of variables is:

$$V = \{x_1(0),\ldots,x_{nx}(0),x_1(1),\ldots,x_{nx}(1),\ldots,x_1(k+1),\ldots,x_{nx}(k+1) \\ \theta_1,\ldots,\theta_{n\theta}\} \tag{8}$$

the set of domains is:

$$D = \{[x_1](0),\ldots,[x_{nx}](0),[x_1](1),\ldots,[x_{nx}](1),\ldots,[x_1](k+1),\ldots,[x_{nx}](k+1) \\ [\theta_1],\ldots,[\theta_{n\theta}]\} \tag{9}$$

and the set of constraints is:

$$
\begin{aligned}
C = \{ & x_1(1) = f_1(x_1(0),\ldots,x_{nx}(0),\theta_1,\ldots,\theta_{n\theta}),\ldots, \\
& x_{nx}(1) = f_{nx}(x_1(0),\ldots,x_{nx}(0),\theta_1,\ldots,\theta_{n\theta}), \\
& \quad \vdots \\
& x_1(k) = f_1(x_1(k-1),\ldots,x_{nx}(k-1),\theta_1,\ldots,\theta_{n\theta}),\ldots, \\
& x_{nx}(k) = f_{nx}(x_1(k-1),\ldots,x_{nx}(k-1),\theta_1,\ldots,\theta_{n\theta}), \\
& x_1(k+1) = f_1(x_1(k),\ldots,x_{nx}(k),\theta_1,\ldots,\theta_{n\theta}),\ldots, \\
& y_{nx}(k+1) = f_{nx}(x_1(k),\ldots,x_{nx}(k),\theta_1,\ldots,\theta_{n\theta}) \\
& \qquad \}
\end{aligned}
\tag{10}
$$

where nx represent the number of system states, $n\theta$ represent the number of system parameters and $y_{nx}(k+1)$ represent the measurement coming from the sensor for system state $x_{nx}(k+1)$.

Also, the consistency can be tested using a time window *L*, instead of solving the CSP with respect to the initial state, this modification reduce the computation time and it allows to be useful in real time. The length of *L* has been studied in the context of interval simulation (Puig, 2003b). When the time window is used, the set of constraints will be:

$$
\begin{aligned}
C = \{ & x_1(k-L) = f_1(x_1(k-L+1),\ldots,x_{nx}(k-L+1),\theta_1,\ldots,\theta_{n\theta}),\ldots, \\
& x_{nx}(k-L) = f_{nx}(x_1(k-L+1),\ldots,x_{nx}(k-L+1),\theta_1,\ldots,\theta_{n\theta}), \\
& \quad \vdots \\
& x_1(k) = f_1(x_1(k-1),\ldots,x_{nx}(k-1),\theta_1,\ldots,\theta_{n\theta}),\ldots, \\
& x_{nx}(k) = f_{nx}(x_1(k-1),\ldots,x_{nx}(k-1),\theta_1,\ldots,\theta_{n\theta}), \\
& x_1(k+1) = f_1(x_1(k),\ldots,x_{nx}(k),\theta_1,\ldots,\theta_{n\theta}),\ldots, \\
& y_{nx}(k+1) = f_{nx}(x_1(k),\ldots,x_{nx}(k),\theta_1,\ldots,\theta_{n\theta}) \\
& \qquad \}
\end{aligned}
\tag{11}
$$

This constraint satisfaction problem is solved using “PROJ2D” (Massa Dao, 2003) solver.

The solution of the above CSP, will provide $\theta_{identified}$ (θ consistent with $y_{nx}(k+1)$).

If $\theta_{identified}$ is empty, it means that there are no system parameter consistent with the measurement coming from the sensor and we can assure that a fault has occurred. However, if $\theta_{identified}$ is not empty, it means that there are parameters consistent with the measurement. $\theta_{identified}$ is a superset of solutions (i.e. an outer solution), then the global consistency must be used in order to assure that the parameters interval indeed are not empty.

3.3 Forward-Backward algorithm for Fault Detection

Finally, the proposed algorithm can be described as follows:

Step 1: Forward test (Vertex simulation)

The interval for the system states at time instant *k+1* $\left[x_{k+1}\right]$ is obtained using vertex simulation (see *Section 3.1*)

If $y_{k+1} \in \left[x_{k+1}\right]$ then NO FAULT

If $y_{k+1} \notin \left[x_{k+1}\right]$ then GOTO Step 2

Step 2: Backward test (Parameter Consistency Check)

$\theta_{identified}$ (i.e. the interval of θ consistent with $y_{nx}\left(k+1\right)$) is obtained applying the algorithm from *Section 3.2.*

$\theta = \theta_{initial} \cap \theta_{identified}$

If $\theta = 0$ then FAULT

If $\theta \neq 0$ then bisections until $\theta = 0$ or θ cannot be reduced more (global consistency is needed) or the algorithm must be extended with another real-time strategy in order to deal with this case.

4. Application

In this section the forward-backward algorithm will be tested using a real benchmark problem, pinpointing the advantages and the drawbacks of the proposed algorithm.

The application example to test the forward-backward algorithm to robust fault detection, deals with an industrial smart actuator consisting of a flow servo-valve driven by a smart positioner, proposed as an FDI benchmark in the European DAMADICS project. The smart actuator consists of a control valve, a pneumatic servomotor and a smart positioner (Bartys, 2002). In this paper, we will focus on the pneumatic servomotor and the electro-pneumatic transducer. The pneumatic servomotor has a non-linear second order dynamic described by:

$$m\frac{d^2X'}{dt^2} = -k_v\frac{dX'}{dt} - k_x(k+X) - F_{vc} + A_eP_s + mg \qquad (12)$$

affected by a hysteresis $X = min(g_h(X', D_b), H_b)$ and the vena contracta force F_{vc} (Bartys, 2002). The electro-pneumatic transducer has a non-linear dynamics described by:

$$\frac{dP_s}{dt} = (P_s + P_a)\left(\frac{1}{m_a}\frac{dm_a}{dt} - \frac{A_e}{V_0 + A_e X}\frac{dX}{dt}\right) \tag{13}$$

$$\text{with: } \frac{dm_a}{dt} = \begin{cases} k_1 CVP\sqrt{P_z - P_s} & if \quad CVP > 0 \\ k_1 CVP\sqrt{P_s} & if \quad CVP \le 0 \end{cases} \tag{14}$$

A simplified non-linear model, without static non-linearities (hysteresis and vena contracta force), for the servomotor that relates X (the servomotor's rod displacement) with CVP (the command presure) will be proposed to be used to detect faults. The air mass variation in the equation (13) will be aproximated with

$$\frac{dm_a}{dt} = k_1 CVP\sqrt{P_z - P_s}\, f_{p1} + k_1 CVP\sqrt{P_s}\, f_{p2} \tag{15}$$

$$\text{where: } f_{p1} = \frac{1}{1 + e^{-bCVP}} \quad \text{and} \quad f_{p2} = 1 - f_{p1} \tag{16}$$

in order to obtain a model that no depends of the input signal sign. In the continous time state space the model is:

$$\begin{aligned}
\frac{dx_1}{dt} &= x_2 \\
\frac{dx_2}{dt} &= -a_{21}x_1 - a_{22}x_2 + a_{23}x_3 + c_2 \\
\frac{dx_3}{dt} &= -a_{32}x_2\frac{x_3 + P_a}{V_0 + A_e x_1} + a_{34}\frac{1}{x_4}\frac{dx_4}{dt}\left(x_3 + P_a\right) \\
\frac{dx_4}{dt} &= k_1 CVP\sqrt{P_z - x_3}\, f_{p1} + k_1 CVP\sqrt{x_3}\, f_{p2}
\end{aligned} \tag{17}$$

where $x_1 = X$ (the rod displacement), $x_2 = \frac{dX}{dt}$, $x_3 = P_s$ (the pressure in the servomotor's chamber), $x_4 = m_a$ (the air mass).

An interval non-linear model for the servomotor that relates X (the servomotor's rod displacement) with CVP (the command presure) will be derived introducing some of the system non-linearities in the structure of the model, and bounding the effect of the rest (hysteresis and vena contracta force) as bounded uncertain parameters.

The non-linear interval model is obtained using the global consistency techniques in fault-free scenario, as it was presented in the *Section 3.2*. Using the consistency technique we check the parameters interval in a time horizont consistents with the input-otput data used for identification.

The consistency identification problem deal with the following discrete time non-linear interval model.

$$
\begin{aligned}
x_1(k+1) &= x_1(k) + \Delta\left(\theta_1 x_2(k) + \theta_2\right) \\
x_2(k+1) &= x_2(k) + \Delta\left(-a_{21}x_1(k) - a_{22}x_2(k) + a_{23}x_3(k) + c_2\right) \\
x_3(k+1) &= x_3(k) + \Delta\left(-a_{32}x_2(k)\frac{x_3(k)+P_a}{V_0 + A_e x_1(k)} + a_{34}\frac{1}{x_4(k)}\frac{x_4(k)-x_4(k-1)}{\Delta}\left(x_3(k)+P_a\right)\right) \\
x_4(k+1) &= x_4(k) + \Delta\left(k_1 CVP\sqrt{P_z - x_3(k)}f_{p1} + k_1 CVP\sqrt{x_3(k)}f_{p2}\right)
\end{aligned}
\tag{18}
$$

where θ_1 and θ_2 are aditional paramaters that was introduced in the model strucure in order to improve the sistem aproximation, Δ is the step size. The bounded uncertain parameters will be $\theta_1 \in \left[\underline{\theta}_1, \overline{\theta}_1\right]$, $\theta_2 \in \left[\underline{\theta}_2, \overline{\theta}_2\right]$, $a_{21} \in \left[\underline{a}_{21}, \overline{a}_{21}\right]$, and $a_{22} \in \left[\underline{a}_{22}, \overline{a}_{22}\right]$. The parameter b from equation (16) was calibrated in order to obtain a good approximation for equation (14).

So, for this identification problem using consistency techniques, the set of constraints will be:

$$
\begin{aligned}
C = \{ & x_1(1) = f_1\left(x_1(0),\ldots,x_4(0),y_1(0),\theta_1,\theta_2,a_{21},a_{22}\right),\ldots, \\
& x_4(1) = f_4\left(x_1(0),\ldots,x_4(0),y_4(0),\theta_1,\theta_2,a_{21},a_{22}\right), \\
& \quad \vdots \\
& x_1(k) = f_1\left(x_1(k-1),\ldots,x_4(k-1),y_1(k-1),\theta_1,\theta_2,a_{21},a_{22}\right),\ldots, \\
& x_4(k) = f_4\left(x_1(k-1),\ldots,x_4(k-1),y_4(k-1),\theta_1,\theta_2,a_{21},a_{22}\right), \\
& x_1(k+1) = f_1\left(x_1(k),\ldots,x_4(k),y_1(k),\theta_1,\theta_2,a_{21},a_{22}\right),\ldots, \\
& x_4(k+1) = f_4\left(x_1(k),\ldots,x_4(k),y_4(k),\theta_1,\theta_2,a_{21},a_{22}\right) \\
& \qquad \}
\end{aligned}
\tag{19}
$$

with the following set of domains:

$$
\begin{aligned}
D = \{ & [x_1](0),\ldots,[x_4](0),[x_1](1),\ldots,[x_4](1),\ldots,[x_1](k+1),\ldots,[x_4](k+1), \\
& [\theta_1]_{initial}, [\theta_2]_{initial}, [a_{21}]_{initial}, [a_{22}]_{initial} \quad \}
\end{aligned} \tag{20}
$$

The consistency problem will provide the bounds for the parameteres $[\theta_1]_{identified}, [\theta_2]_{identified}, [a_{21}]_{identified}, [a_{22}]_{identified}$ so that in fault-free scenario the model envelopes will contain the sistem output coming from the sensores.

This non-linear interval model suffer of the wrapping effect due to not fulfil the isotony property (Cugueró, 2002). For this non-isotonic systems the exact envelopes computation is time-consuming due to the global optimisation problem that must be solved, the one step-ahead algorithms fails very quickly providing an instable interval simulation and the vertex simulation provide an inner solution.

Finally, this non-linear interval model will be used and the forward-backward algorithm will be applied in order to detect faults.

In this example, a fault in the pneumatic servomotor is introduced. The fault consists in servomotor's diaphragm perforation caused by fatigue of diaphragm material (named f_{10} in the DAMADICS benchmark). In the present experiments one fault scenario will be used corresponding to the abrupt medium size (Bartys, 2002). The fault appears at time instant *t=200* s. Due to integration reasons (Euler's integration algorithm) the step size was chosen $\Delta = 0.0005\ s$.

In the figure 1 results applying forward-backward algorithm is presented. From initial time *t=0* to *t=200* s forward test (**Step 1** of the algorithm) was used. The pneumatic servomotor non-linear behaviour is better approximated using a non-linear interval model instead of the linear one. Due to this, the envelopes are less conservative in the non-linear case than in the linear one (Puig, 2003a) and the system output falls outside the envelopes very quickly when the fault has occurred.

When the system output falls outside the envelopes, the backward test (**Step 2** of the algorithm) will be used. This test performed an empty interval parameter consistent with the measure and we can assure that the fault occurred.

This algorithm can provide an undecided zone when the fault is not permanent and the fault size is very small (as we can see in the figure 2). In this case the measurement is outside the inner solution but very close to inner solution and can belong between the inner and unknown exact solution. In this case the local consistency test can provide a non empty parameters interval consistent with the measurement, i.e. the measurement does not invalidate the interval model. In this situation, due to the local consistency used for backward test (superset of solutions) we cannot assure that there is a fault-free scenario or not.

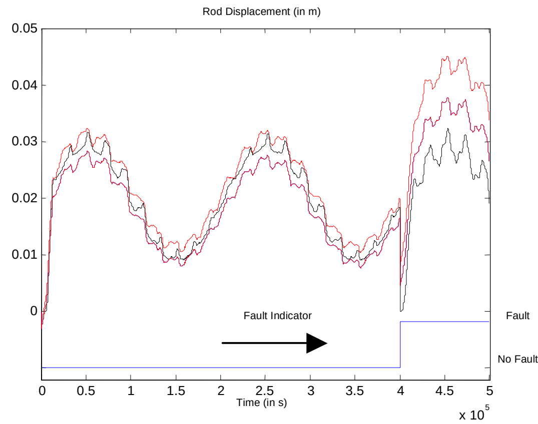


Fig. 1 Fault detection (f_{10} medium size)

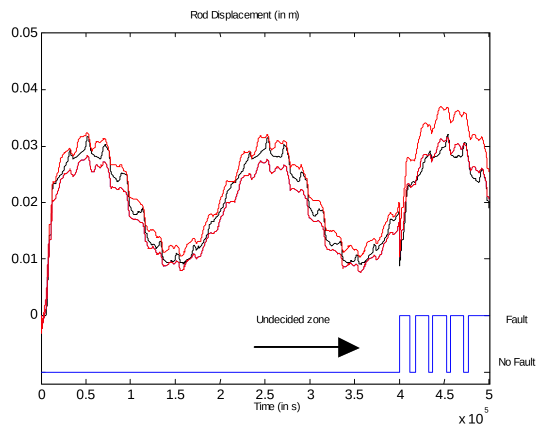


Fig. 2 Undecided zone for a small size fault

5. Conclusions

Considering the problems that appear in interval simulation/observation using regions or real trajectories, a new algorithm for fault detection was proposed. This algorithm uses a vertex simulation to compute the envelopes due to its computational simplicity. The consistency test must be used only when a measurement coming from the sensor falls outside the inner solution. When this measurement belongs to the region between the inner solution and the exact solution (unknown) the consistency test provides an interval $\theta_{identified}$ nonempty and we cannot decide that this measurement represents a fault or a normal situation. When the measurement is outside the exact envelopes, the consistency test provides very quickly an empty interval $\theta_{identified}$ assuring that a fault occurred.

Due to incompleteness of the solution, the forward test based on vertex simulation only can assure that if the measurements coming from the sensors falls inside the inner solution, then the system work properly. When a measurement falls outside we cannot conclude anything about that. Using the consistency test (outer solution) for a measurement we can assure that a fault occurred only when $\theta_{identified}$ is an empty set. There is an undecided zone between the inner solution and the exact solution (unknown). In this case we only can say that a measurement represents a possible fault.

In conclusion this forward-backward algorithm is developed in order to be applied in fault detection applications where real-time operation is needed.

As a future work we want to minimize as much is possible this undecided zone, and to combine the forward-backward algorithm with an intelligent test in order to decide about the measurements that belongs to the undecided zone.

References

1. **(Bartys, 2002)** Bartys, M.Z. "Specification of actuators intended to use for benchmark definition". http://diag.mchtr.pw.edu.pl/damadics/
2. **(Chen, 1999)** Chen J. and R.J. Patton (1999). "Robust Model-Based Fault Diagnosis for Dynamic Systems". Kluwer Academic Publishers.
3. **(Cugueró, 2002)** Cugueró, P., Puig, V., Saludes, J., Escobet, T. "A Class of Uncertain Linear Interval Models for which a Set Based Robust Simulation can be Reduced to Few Pointwise Simulations". In Proceedings of Conference on Decision and Control 2002 (CDC'02). Las Vegas. USA.
4. **(Gertler, 1998)** Gertler, J.J. "Fault Detection and Diagnosis in Engineering Systems". Marcel Dekker
5. **(Jaulin, 2002)** Jaulin, L., "Consistency techniques for the localization of a satellite". 1st International Workshop on Global Constrained Optimization and Constraint Satisfaction (Cocos'02), Valbonne - Sophia Antipolis, France. October 2-4, Nice.
6. **(Kolev, 1993)** Kolev, L.V. "Interval Methods for Circuit Analysis". Singapore. World Scientific.
7. **(Massa Dao, 2003)** "Proj2D Solver". http://www.istia.univ-angers.fr/~dao/
8. **(Nickel,1985)** Nickel, K.. "How to fight the wrapping effect". In K. Nickel ed. "Interval Analysis 1985". Lecture Notes in Computer Science, No. 212, pp. 121-132. Springer-Verlag.
9. **(Puig, 2002)** Puig, V., Quevedo, J., Escobet, T., De las Heras, S. "Robust Fault Detection Approaches using Interval Models". IFAC World Congress (b'02). Barcelona. Spain.
10. **(Puig, 2003a)** Puig, V., Quevedo, J., Escobet, T., Stancu, A. "Passive Robust Fault Detection using Linear Interval Observers…". IFAC Safe Process, 2003. Washington. USA.
11. **(Puig, 2003b)** Puig, V., Saludes, J., Quevedo, J. "Worst-Case Simulation of Discrete Linear Time-Invariant Dynamic Systems", Reliable Computing 9(4): 251-290, Aug. 2003.