Moving Horizon State Estimation of Hybrid Systems. Application to Fault Detection of Sensors of a Steam Generator

JEAN THOMAS (1), DIDIER DUMUR (1), JEAN BUISSON (2), C.M. FALINOWER (3), P. BENDOTTI (3)

(1) Supélec F 91192 Gif sur Yvette cedex – Phone : +33 (0)1 69 85 13 79
(2) Supélec F 35 511 Cesson-Sévigné cedex – Phone : +33 (0)2 99 84 45 42
Email : {jean.thomas, didier.dumur, jean.buisson}@supelec.fr
(3) EDF R&D F 78401 Chatou – Phone : +33 (0)1 30 87 70 77
Email : {clement-marc.falinower,pascale.bendotti}@edf.fr

Abstract— The Mixed Logical Dynamical (MLD) formalism has proved to be an efficient modeling framework for hybrid systems described by dynamics, logic and constraints. Furthermore, it allows formulating and solving practical problems such as moving horizon control and state estimation. Starting from the general framework of state estimation, this paper presents a modified version adapted to fault detection of hybrid systems under the MLD formalism. A decomposition into several sub-problems decreases the computation time for a faster real time implementation. This strategy is applied in simulation to fault detection of sensors of a steam generator benchmark.

Keywords— Hybrid systems, Mixed logical dynamical systems, State estimation, Fault detection, Steam generator.

I. INTRODUCTION

Many physical processes can be described as hybrid systems, including both continuous and discrete variables, discrete variables coming from parts described by logic such as for example on/off switches or valves. Various approaches have been proposed for modeling hybrid systems [7], like Automata model, Petri nets model, Linear Complementary (LC) model. It was shown that moving logical relations into linear constraints on integer variables provides a global modeling framework called Mixed Logical Dynamical (MLD) formalism [1]. It allows describing a large number of important classes of hybrid systems, such as piecewise linear systems, systems with mixed continuous/discrete inputs and states ...

Problems related to hybrid systems, such as control, stability, state estimation and fault detection, become an interesting field of investigation. In fact the state estimation and sensors fault detection issue is widely addressed in the literature, in the case of non hybrid systems, e.g. [9], [14] and [12]. Considering hybrid systems, a moving horizon criteria for state estimation under the MLD formalism was presented in [2], with extension to fault detection. Other approaches of fault detection of hybrid systems can also be found, for example in [8], [10] and [11].

Starting from the strategy developed in [2], this paper presents a modified version mostly dedicated to fault detection of sensors of hybrid systems modeled with the MLD formalism. The deduced cost function is split into several independent sub-criteria, each one related to a particular subset of sensors. This enables to decrease the calculation time of the estimation part in a significant way for real time implementation. An application to fault detections of sensors of a steam generator is presented which validates the proposed structure.

The paper is organized as follows. Section II presents a brief description of the MLD systems. The moving horizon state estimation strategy for hybrid systems and its corresponding cost function are developed in section III. Section IV examines the steam generator benchmark and its specification within the fault detection framework. Finally, section V presents the application of the estimation strategy to the fault detection of sensors of the steam generator.

II. MLD MODEL

The Mixed Logical Dynamical (MLD) model permits the description of various classes of hybrid systems, like linear hybrid systems, constrained linear systems, sequential logical systems (finite state machines, automata), some classes of discrete event systems, and non-linear dynamic systems, where non-linearities can be expressed through logical combination. The MLD model describes the system by linear dynamic equations subject to linear inequalities involving both real and integer variables, under the following form (see [1] for more details):

\[
\begin{align*}
\dot{x}(t+1) &= Ax(t) + Bu(t) + B_2\delta(t) + B_3z(t) \\
y(t) &= Cx(t) + Du(t) + D_2\delta(t) + D_3z(t) \\
E_2\delta(t) + E_3z(t) &\leq E_1u(t) + E_4x(t) + E_5
\end{align*}
\]

where:

\[
x = \begin{bmatrix} x_c \\ x_l \end{bmatrix} \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}, \quad u = \begin{bmatrix} u_c \\ u_l \end{bmatrix} \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}, \\
y = \begin{bmatrix} y_c \\ y_l \end{bmatrix} \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}, \quad \delta \in \{0,1\}^{q}, \quad z \in \mathbb{R}^{r_c}
\]

are respectively the vectors of continuous and binary states of the system, of continuous and binary (on/off) control inputs, of output signals, of auxiliary binary and continuous variables. The auxiliary variables are introduced when translating propositional logic into linear inequalities as described in figure 1. A MLD model Eq. 1 thus represents logical relations and interaction between continuous and logical variables by mixed integer linear inequalities.
the following estimate evolution:

\[
\mathbf{x}(t - T|t) = \hat{\mathbf{x}}(t - T|t - 1) + \Delta \mathbf{x}(t)
\]

\[
\hat{\mathbf{x}}(t + k|t) = \mathbf{A}\hat{\mathbf{x}}(t + k|t) + \mathbf{B}_1\mathbf{u}(t + k|t) + \mathbf{B}_2\hat{\mathbf{z}}(t + k|t) +
\]

\[
+ \mathbf{B}_3\hat{\mathbf{z}}(t + k|t) + \mathbf{B}_6\phi(t + k|t) + \xi(t + k|t)
\]

\[
\hat{\mathbf{z}}(t + k|t) = \mathbf{C}\hat{\mathbf{x}}(t + k|t) + \mathbf{D}_1\mathbf{u}(t + k|t) + \mathbf{D}_2\hat{\mathbf{z}}(t + k|t) +
\]

\[
+ \mathbf{D}_4\hat{\mathbf{z}}(t + k|t) + \mathbf{D}_6\phi(t + k|t) + \zeta(t + k|t)
\]

\[
\mathbf{E}_2\hat{\mathbf{z}}(t + k|t) + \mathbf{E}_3\hat{\mathbf{z}}(t + k|t) \leq \mathbf{E}_4\hat{\mathbf{x}}(t + k|t) + \mathbf{E}_5\mathbf{u}(t + k|t) +
\]

\[
+ \mathbf{E}_5 + \mathbf{E}_6\phi(t + k|t) \quad \text{with} \quad k \leq 0
\]

The optimization variables at time \( t \) are organized in the following vector:

\[
\mathbf{z}_t = [\Delta \mathbf{x}(t), \hat{\mathbf{z}}(t - T|t), \hat{\mathbf{x}}(t - T|t), \hat{\mathbf{z}}(t - T|t), \cdots, \hat{\mathbf{x}}(t - |T - 1|), \hat{\mathbf{z}}(t - |T - 1|), \cdots, \hat{\mathbf{x}}(t - 1), \hat{\mathbf{z}}(t - 1)]
\]

\[
\mathbf{\Phi}(t - T|t) = [\Delta \mathbf{x}(t), \hat{\mathbf{z}}(t - T|t), \hat{\mathbf{x}}(t - T|t), \hat{\mathbf{z}}(t - T|t), \cdots, \hat{\mathbf{x}}(t - 1), \hat{\mathbf{z}}(t - 1)]
\]

\[
\mathbf{X}(t - 1) = [\hat{\mathbf{x}}(t - T|t - 1), \hat{\mathbf{z}}(t - T + |T - 1|), \cdots, \hat{\mathbf{z}}(t - T|t - 1), \hat{\mathbf{x}}(t - T + |T - 1|), \cdots, \hat{\mathbf{x}}(t - 1), \hat{\mathbf{z}}(t - 1)]
\]

Consider at time \( t \) the following estimate evolution:

\[
\hat{\mathbf{x}}(t + k|t) = \mathbf{A}\hat{\mathbf{x}}(t + k|t) + \mathbf{B}_1\mathbf{u}(t + k|t) + \mathbf{B}_2\hat{\mathbf{z}}(t + k|t) +
\]

\[
+ \mathbf{B}_3\hat{\mathbf{z}}(t + k|t) + \mathbf{B}_6\phi(t + k|t) + \xi(t + k|t)
\]

\[
\hat{\mathbf{z}}(t + k|t) = \mathbf{C}\hat{\mathbf{x}}(t + k|t) + \mathbf{D}_1\mathbf{u}(t + k|t) + \mathbf{D}_2\hat{\mathbf{z}}(t + k|t) +
\]

\[
+ \mathbf{D}_4\hat{\mathbf{z}}(t + k|t) + \mathbf{D}_6\phi(t + k|t) + \zeta(t + k|t)
\]

\[
\mathbf{E}_2\hat{\mathbf{z}}(t + k|t) + \mathbf{E}_3\hat{\mathbf{z}}(t + k|t) \leq \mathbf{E}_4\hat{\mathbf{x}}(t + k|t) + \mathbf{E}_5\mathbf{u}(t + k|t) +
\]

\[
+ \mathbf{E}_5 + \mathbf{E}_6\phi(t + k|t) \quad \text{with} \quad k \leq 0
\]

The optimization variables at time \( t \) are organized in the following vector:

\[
\mathbf{z}_t = [\Delta \mathbf{x}(t), \hat{\mathbf{z}}(t - T|t), \hat{\mathbf{x}}(t - T|t), \hat{\mathbf{z}}(t - T|t), \cdots, \hat{\mathbf{x}}(t - 1), \hat{\mathbf{z}}(t - 1)]
\]

\[
\mathbf{\Phi}(t - T|t) = [\Delta \mathbf{x}(t), \hat{\mathbf{z}}(t - T|t), \hat{\mathbf{x}}(t - T|t), \hat{\mathbf{z}}(t - T|t), \cdots, \hat{\mathbf{x}}(t - 1), \hat{\mathbf{z}}(t - 1)]
\]

\[
\mathbf{X}(t - 1) = [\hat{\mathbf{x}}(t - T|t - 1), \hat{\mathbf{z}}(t - T + |T - 1|), \cdots, \hat{\mathbf{z}}(t - T|t - 1), \hat{\mathbf{x}}(t - T + |T - 1|), \cdots, \hat{\mathbf{x}}(t - 1), \hat{\mathbf{z}}(t - 1)]
\]
\[ y = y_s + z \text{ with } z = \phi g \]  
\text{(10)}

where \( g \) is a continuous function corresponding to the fault profile, \( y \) and \( y_s \) are respectively the sensor output and the measured variable.

Let introduce the following notations:
- \( p = p_c + p_f \) is the total number of measured variables (i.e. system outputs)
- \( q_i/j = 1, \ldots , p \) represents the number of sensors measuring the same \( i^{th} \) output \( y_{si} \)
- \( h_{ij}/j = 1, \ldots , p; i = 1, \ldots , q_i \) represents the number of faults which can influence the \( j^{th} \) sensor measuring the \( i^{th} \) output \( y_{si} \)
- \( h_i = 1, \ldots , p \) represents the number of faults which can influence all the sensors measuring the \( i^{th} \) output \( y_{si} \)

Based on the framework of Eq. 10 and the previous definitions, the total number of additional auxiliary binary variables needed to model the faults appearing on all the sensors is:

\[ L_d = \sum_{i=1}^{p} \sum_{j=1}^{q_i} h_{ij} = \sum_{i=1}^{p} h_i \]  
\text{(11)}

Assuming that only one fault could happen at the same instant on the same sensor \( j \) of a particular measured output \( i \), the relation between all the additional auxiliary binary variables is:

\[ \sum_{k=1}^{g} \phi_{jk}^i \leq 1 \quad j = 1, \ldots , q_i; i = 1, \ldots , p \]  
\text{(12)}

The expression of the \( j^{th} \) sensor output measuring the \( i^{th} \) output is given by:

\[ y_{ij} = y_{si} + \sum_{k=1}^{h_i} \phi_{jk}^i \text{ under Eq. 12} \]  
\text{(13)}

Eq. 12 corresponds in fact to an additional constraint introduced within the MLDF model Eq. 2.

Finally, the total number of binary optimization variables which are present in Eq. 8 is:

\[ L = (T + 1)(r_l + L_d) + n_f \]  
\text{(14)}

where \( n_f \) is the number of logical state variables and \( r_l \) is the number of auxiliary logical variables of the MLD model without modeling the faults, which appear during the minimization.

In the worst case, the optimization time of the MIQP optimization problem, Eq. 8, increases exponentially with the number of binary optimization variables [13]. It must be noticed that \( L_d \) auxiliary continuous variables are added too, but with insignificant effect on the computation time compared to the added binary variables.

To reduce the computation time for real time implementation purposes, we consider now partitioning the previous optimization problem into \( p \) independent MIQPs, each one dedicated to state estimation and fault detection of the sensors of a measured output \( y_{sj} \). This separation is in fact possible if and only if the sensors acting on a particular output are not influenced by the sensors of the remaining outputs.

Even if the problem nature remains exponential, this yields solving \( p \) simpler Branch and Bound (B&B) sub-trees instead of a global more complex tree. Of course each of the \( p \) optimization problems uses the same control data \( U(t) \) and the MLDF model of the system, Eq. 2, with for the \( i^{th} \) one:

\[ \phi(t) = \left[ \phi_{jk}^i \right]_{j=1, \ldots , q_i; k=1, \ldots , h} \]  
\text{(15)}

With this technique, it may happen that some state systems become unobservable, since only one output is considered in a sub-problem. In this case, states are reconstructed at each iteration using only the sensors outputs that are not affected by any fault.

As previously mentioned, this technique considerably reduces the optimization time, since solving one tree with \( L_d \) added optimization variables takes much more time than solving \( p \) trees, each one with only \( L_{di} \) added binary variables, with:

\[ L_d = \sum_{i=1}^{p} L_{di} \]  
\text{(16)}

Finally, as the fault values \( g \) are also unknown values, additional states may be included in the state vector \( x \) of the MLDF model, so that an estimation of these values can be performed if required.

IV. DESCRIPTION OF THE STEAM GENERATOR BENCHMARK

The main objective in controlling a PWR (Pressurized Water Reactor) is to provide the commanded power while respecting certain physical constraints. The pressurized water in the primary circuit transmits the heat generated by the nuclear reaction to the steam generator (SG). In the SG, water of secondary circuit is converted to steam, which drives a turbo-alternator to generate electricity.

The main goal of the control strategy aims at tracking a reference setpoint and maintaining the SG water level within permitted limits, even with changes in the steam flow-rate \( Q_e \) (connected to changes in the power demand) considered as a disturbance, by acting on the \( Q_{ec} \) control signal, corresponding to the input of the valve delivering the feed-water flow-rate \( Q_e \).

A. General description

The SG is consequently the interface between the primary and secondary circuits. As the secondary fluid heats up due to the heat exchange with the primary fluid, it turns into a two-phase fluid. Due to the two-phase nature of the steam-water mixture in the SG, the water level is not a well defined quantity. Two types of water level measurement are available: the narrow range level \( N_{gE} \) reflecting the “mixture level” and the wide range level \( N_{gI} \) reflecting the mass of water in the SG.

According to this description, the SG can be modeled as a two inputs \( (Q_{ec}, Q_v) \), four measured outputs \( (Q_e, Q_v, N_{ge}, N_{gi}) \) system. Moreover, two additional difficulties arise. First, to cope with potential sensor failures, sensor redundancy is imposed. Then, at low powers \( (<50\%P_n) \), where \( P_n \) corresponds to the nominal power), as flow-rate sensors are not reliable, estimates values of \( Q_e \) and \( Q_v \) are used instead of measured values.
In order to provide the controller with reliable measurements, the following developments will now focus on the selection of the appropriate information, using the fault detection theory developed in Section III.

B. Simplified dynamical model

Detailed theoretical models of the SG based on fundamental conservation equations and thermodynamic principles are too complex; therefore a simplified model is used for control purposes, according to the following state space form:

\[
x(t) = \begin{pmatrix}
\alpha & 0 & 0 & 0 & 0 \\
\tau_e^{-1} & 0 & 0 & 0 & 0 \\
0 & -\tau_e & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -\tau_g^{-1}
\end{pmatrix}
\begin{pmatrix}
x(t) \\
\end{pmatrix}
+ \begin{pmatrix}
0 \\
-\tau_e^{-1} \\
0 \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
u(t) \\
d(t)
\end{pmatrix}
\]

\[
y(t) = \begin{pmatrix}
\beta_g \tau_g & 0 & 0 & 0 & 0 \\
0 & \beta_0 & \beta_0 & 1 & 0 \\
0 & 0 & \tau_n \tau_l^{-1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
x(t) + d(t) \\
\end{pmatrix}
\]

with \( y(t) = \begin{pmatrix} N_{ge,1} & N_{ge,2} & Q_{e,1} & Q_{e,2} \end{pmatrix}^T \) the measured outputs, \( u(t) = Q_{ec} \) the control signal corresponding to the feed-water flow-rate reference, \( d(t) = Q_v(t) \) the disturbance.

This model (see [5] and [6] for more details) exhibits the main characteristics of the plant behavior, i.e. the mass capacity effect of the SG and the “swell and shrink” effect inducing a non minimum phase behavior, and also includes a first order term for the feed-water actuator. All these dynamics change with the operating power. Parameters values at 80% \( P_n \) and 10% \( P_n \) are given in Table 1.

<table>
<thead>
<tr>
<th>Parameters \ Power</th>
<th>80% ( P_n )</th>
<th>10% ( P_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_e )</td>
<td>5.435</td>
<td>2.433</td>
</tr>
<tr>
<td>( \tau_n )</td>
<td>13.889</td>
<td>21.739</td>
</tr>
<tr>
<td>( \alpha_i )</td>
<td>0.983</td>
<td>0.171</td>
</tr>
<tr>
<td>( \alpha_0 )</td>
<td>0.811</td>
<td>0.025</td>
</tr>
<tr>
<td>( \tau_g )</td>
<td>4.167</td>
<td>2.410</td>
</tr>
<tr>
<td>( \beta_g )</td>
<td>1.325</td>
<td>0.990</td>
</tr>
<tr>
<td>( \beta_0 )</td>
<td>0.106</td>
<td>0.068</td>
</tr>
<tr>
<td>( \tau_f )</td>
<td>694.45</td>
<td>701.258</td>
</tr>
</tbody>
</table>

Table 1: Parameters values at 80% \( P_n \) and 10% \( P_n \).

This model used for control design will also be considered in the framework of fault detection of sensors.

C. Specification for fault detection

The reliability of the SG water level control is assured in practice by sensor redundancy. In our case, the available measurements are the following:

- Two sensors \( Q_{e,1} \) and \( Q_{e,2} \) measure the feed-water flow-rate \( Q_e \) and are used at high powers (> 50% \( P_n \)).
- Two sensors \( Q_{v,1} \) and \( Q_{v,2} \) measure the steam flow-rate \( Q_v \), and are used at low powers (> 50% \( P_n \)).
- Three sensors \( N_{ge,1}, N_{ge,2} \) and \( N_{ge,3} \) measure the narrow range level.
- One sensor \( N_{gf,1} \) measures the wide range level.

As mentioned in the previous section, estimates values of \( Q_e \) and \( Q_v \) are used at low powers, their estimation strategy based on first stage pressure and control valve position is not considered in the following.

The following failure cases will be considered:

- “Straight” loss or fast drift of measurement within sensor admissible range.
- Slow drift of the measurement.

Furthermore, in normal operation, any measurement is subject to noise and offset, meaning that the different sensors used for the same physical measurement always differ from one another. Finally, it is assumed that only one fault could happen at the same instant on the same sensor.

V. SIMULATION RESULTS

Simulations are performed with an Euler discretization of the previous dynamical model, for a sampling period of 1 second.

A. Fault modeling within the MLD framework

Figure 2 presents, according to the theory of Section 3, an illustrative example of modeling the faults effect in the MLDF form. Let consider the fast and slow drift faults that may appear on two sensors measuring the particular SG output \( y_1 \). Variables \( g_1 \) and \( g_2 \) represents the fast and slow drift fault values respectively, \( y_1 \) and \( y_2 \) the sensors outputs. With the developed formalism, \( \phi_1 = 1 \) when the sensor 1 is influenced by the fast drift \( g_1 \).

![Fig. 2. Fault structures influencing the sensors.](image)

The example of Figure 2 is translated into the MLDF form through the following equations:

\[
\begin{align*}
y_1 &= y_2 + z_1 + z_2 \\
y_2 &= y_3 + z_3 + z_4
\end{align*}
\]

\[
\begin{align*}
z_1 &= \phi_1 g_1 & (i.e. \text{ if } \phi_1 = 1 \rightarrow z_1 = g_1) \\
z_2 &= \phi_2 g_2 & (i.e. \text{ if } \phi_2 = 1 \rightarrow z_2 = g_2) \\
z_3 &= \phi_3 g_1 & (i.e. \text{ if } \phi_3 = 1 \rightarrow z_3 = g_1) \\
z_4 &= \phi_4 g_2 & (i.e. \text{ if } \phi_4 = 1 \rightarrow z_4 = g_2)
\end{align*}
\]
where \( \phi_1, \phi_2, \phi_3, \phi_4 \in [0,1] \) are additional binary variables.

The benchmark considers only one fault occurring on a same output, leading to the following relation:
\[
(\phi_1 + \phi_2 + \phi_3 + \phi_4) \leq 1
\]  
(19)

However, the developed theory can deal with several faults occurring on the same output.

Consequently, as the SG benchmark includes eight sensors, each of them subject to two faults, 16 additional binary variables are required to model all the fault occurrence possibilities.

The optimization problem has been divided into four sub-problems with four B&B trees for the four split physical measured quantities \( N_{ge}, N_{gl}, Q_v, Q_e \), and as previously detailed, the resulting optimization solving time is much smaller than the global optimization problem.

**B. Case I: detection of fast drift on the \( N_{ge} \) sensors**

A simulation of the SG benchmark has been performed with the features: first a +5% step change on \( Q_e \) at high power (85% \( P_e \)), and a fast drift on one of the \( N_{ge} \) sensors. The estimation and fault detection result is shown in Figure 3. The hybrid strategy succeeds in detecting the fault occurrence on \( N_{ge3} \) without any delay. The recomputed value for \( N_{ge} \) is finally the average value of data coming from sensors that are not influenced by any fault. In the proposed case, the recomputed value is the average value of the three sensors up to \( t = 260s \) (the time instant where the fault on \( N_{ge3} \) was detected), then it becomes the average value of the two remaining sensors \( N_{ge1} \) and \( N_{ge2} \). This recomputed value is shown on the lowest part of figure 3.

![Fig. 3: Fault detection results for a fast drift fault on \( N_{ge} \).](image)

**C. Case II: detection of slow drift on the \( N_{ge} \) sensors**

A simulation of the SG benchmark has been performed with the features: first a +5% step change on \( Q_e \) at low power (10% \( P_e \)), and a slow drift on one of the \( N_{ge} \) sensors. The estimation and fault detection result is shown in Figure 4.

The hybrid strategy succeeds in detecting the fault occurrence on \( N_{ge1} \) with a small acceptable delay (320 seconds), depending on the fault magnitude, the weighting factors of Eq. 7 and the estimation horizon \( T \). These two last impacts are detailed in the next paragraphs. The \( N_{ge} \) value is finally recomputed in the same way as in previous part and shown on the lowest part of figure 4.

![Fig. 4: Fault detection results for a slow drift fault on \( N_{ge} \).](image)

**D. Influence of the estimation horizon \( T \)**

Increasing the estimation horizon provides a better identification of the fault type (i.e. fast or small drift), and a smaller delay for the detection of the slow drift fault occurrence, but requires more computation time. The previous results of Figures 3 and 4 are obtained with an estimation horizon \( T = 3 \), and in this case the fault detection strategy gives a correct identification of the type of fault with a 80% rate. With \( T = 1 \), the optimization program is much faster; for example the maximum optimization solving time for case I of Figure 3 is 7 seconds with \( T = 1 \) while it is 200 seconds with \( T = 3 \), also for case II of Figure 4, it is 8 seconds with \( T = 1 \) and 300 seconds with \( T = 3 \) (using the mipq.m Matlab code [3] on a 1.8 MHz Pc with 256 kram). With \( T = 1 \), the hybrid strategy also succeeds in detecting the fault occurrence, but with an increased delay in the case of a slow drift fault, and a correct identification of the type of the fault with a poor rate. Such an example of the estimation horizon effect is shown Figure 5, in the case of a slow drift fault on \( N_{gl} \) sensor, with a +5% step change on \( Q_e \) at high power (5% \( P_e \)). It can be seen that the fault occurrence on \( N_{gl} \) is detected later in case \( T = 3 \) (370 sec.) than that with \( T = 6 \) (240 sec.).

![Fig. 5: Fault detection results for a slow drift fault on \( N_{gl} \) with different values of the estimation horizon.](image)
E. Weighting factors impact

The choice of the $Q_i$ weighting factors of Eq. 7 influences the delay of the detection of the slow drift fault occurrence, especially $Q_3$, $Q_6$ and $Q_{10}$. An example of the weighting factors effect is shown Figure 6, where simulation was realized with different values of these three weighting factors (case (a) $Q_3 = 2000, Q_6 = 100, Q_{10} = 3000$, case (b) $Q_3 = 3000, Q_6 = 500, Q_{10} = 7000$), in the case of a slow drift fault on one of the $Q_i$ sensors at high power ($85\% P_{in}$).

Lower values of those weighting factors make the estimation more sensitive and enable the fault detection occurrence with a reduced delay (130 seconds for case (a) of Figure 6 instead of 250 seconds for case (b) of Figure 6). But very small values must be avoided, making the system too sensitive, even detecting wrong features such as noise impact and biased signals. Trade-off between the fault detection delay and the false fault detection has thus to be made.

![Graph 6: Fault detection results for a slow drift fault on Q.](image)

VI. CONCLUSION

This paper presents a strategy dedicated to state estimation and fault detection of hybrid systems under the MLD formalism. The decomposition into several sub-problems reduces the computation burden and is of most interest for real time implementation. The developed structure has been successfully implemented to detect the sensors faults of a steam generator benchmark, either the fast drift fault without any delay, and the slow drift fault with a small delay; this could be explained by the fact that the fast drift fault value is large enough to be detected without any delay, while the slow drift fault starts with a small value and can only be detected when its value becomes significant. An analysis of the tuning parameters has shown that this delay and the identification of the fault type both depend mainly on the choice of the weighting factors and the estimation horizon.

Future work may consider looking for a procedure which would help achieving the best compromise between the fault detection occurrence, the estimation of the adequate type of fault and the calculation time, when choosing the weighting factors and the estimation horizon. Another research field based on the SG benchmark consists in taking into account the recomputed outputs resulting of the state estimation and fault detection program as an input to a water level hybrid controller, in order to get a consistent approach (both hybrid fault detection and hybrid control).

VII. REFERENCES


