FITBOX – A Fault Isolation Toolbox

Pedro Casau * **Paulo Rosa** ** **Carlos Silvestre** *,***

* *Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal*
** *Deimos Engenharia, Lisbon, Portugal*
*** *Department of Electrical and Computer Engineering, Faculty of Science and Technology of the University of Macau, China*

Abstract: In this paper, we present the theoretical background for the implementation of FITBOX – a new freely available fault isolation toolbox for MATLAB that makes use of novel set-membership methods for Fault Detection and Isolation (FDI). We apply the proposed methods to the FDI of a wind turbine and evaluate their performance in a simulation setting.

1. INTRODUCTION

The subject of Fault Detection and Isolation (FDI) has been the focus of intensive research efforts over the last few decades, namely, on the development of residual-based strategies for linear systems, in an effort pioneered by Beard (1971) and Jones (1973). These strategies consist in the design of an output signal – *residual* – that is non-zero only if a given fault signal is also different from zero. In practice, due to the presence of sensor noise and unmodeled dynamics, this is hardly ever the case, and thus a decision threshold has to be computed. If the decision threshold is not sufficiently large, then the susceptibility of the FDI system to false detections increases. If, however, it is set to a large value, then a fault occurrence might go unnoticed Frank (1990). This limitation has fueled a number of contributions on robust residual-based strategies, namely the works by Frank and Ding (1994), Gertler and Kunwer (1995) and De Persis and Isidori (2001). For further information, on residual-based FDI strategies, the reader is referred to Chen and Patton (2012), and, for an alternative FDI toolbox using residual-based methods, the reader is referred to Varga (2006).

More recently, considerable effort has been devoted to the development of FDI strategies based on set-membership methods Puig (2010), which encompass parameter estimation methods, such as Jauberthie et al. (2013) and Ingimundarson et al. (2009), as well state estimation methods, such as Combastel and Raka (2009) and Rosa (2011). In FITBOX, we follow closely the work of Rosa (2011) which is described below.

FITBOX is suitable for the fault detection and isolation of system that can be described by a discrete-time Linear Parameter-Varying (LPV) model of the form

$$\begin{aligned} x(k+1) &= A(\phi(k))x(k) + B(\phi(k))u(k) + L(\phi(k))w(k) \\ y(k) &= C(\phi(k))x(k) + D(\phi(k))u(k) + N(\phi(k))w(k), \end{aligned} \tag{1}$$

where $x(k) \in \mathbb{R}^{n_x}$ denotes the state of the system at time $k \in \mathbb{N}$ for some $n_x \in \mathbb{N}$, $y(k) \in \mathbb{R}^{n_y}$ denotes the output at time $k \in \mathbb{N}$ for some $n_y \in \mathbb{N}$, $w(k) \in \mathbb{R}^{n_w}$ denotes an unknown disturbance signal at time $k \in \mathbb{N}$ for some $n_w \in \mathbb{N}$, $u(k) \in \mathbb{R}^{n_u}$ denotes a known input at time $k \in \mathbb{N}$ for some $n_u \in \mathbb{N}$, $\phi(k) \in \mathbb{R}^{\ell}$ represents a vector of parameters whose evolution in time $k \in \mathbb{N}$ is typically unknown during design. Moreover, the operators $A : \mathbb{R}^{\ell} \to \mathbb{R}^{n_x \times n_x}$, $B : \mathbb{R}^{\ell} \to \mathbb{R}^{n_x \times n_u}$, $L : \mathbb{R}^{\ell} \to \mathbb{R}^{n_x \times n_w}$, $C : \mathbb{R}^{\ell} \to \mathbb{R}^{n_y \times n_x}$, $D : \mathbb{R}^{\ell} \to \mathbb{R}^{n_y \times n_u}$ and $N : \mathbb{R}^{\ell} \to \mathbb{R}^{n_y \times n_w}$ depend on this vector of parameters.

The implementation of the adopted FDI strategy is separated into two classes of objects: the Set-Valued Observer (SVO) class and the fault isolation module. The former class has the set of methods discussed in Section 3, which implement the basic SVO functionality. The latter class is described in Section 4 and makes use of the SVOs to determine whether the system is undergoing a faulty behavior or not. The reader may download the FITBOX source code and the implementation of the examples given in this paper from Casau et al. (2014a). For alternative solutions to the ones provided in FITBOX, the reader is referred to Kurzhanskiy and Varaiya (2006) and Hargreaves (2002)

2. PRELIMINARIES

$\mathbb{R}^n$ denotes the n-dimensional Euclidean space, equipped with the Euclidean inner product $\langle u, v\rangle := u^\top v$, for each $u, v \in \mathbb{R}^n$, and the vector norm $|v| := \sqrt{\langle v, v\rangle}$ for each $v \in \mathbb{R}^n$. Given a vector $v \in \mathbb{R}^n$, each of its components is denoted by v_i with $i \in \{1, 2, \ldots, n\}$. Given $u, v \in \mathbb{R}^n$, $u \preceq v$ if $u_i \leq v_i$ for each $i \in \{1, 2, \ldots, n\}$. Similarly, $u \succeq v$ if $u_i \geq v_i$ for each $i \in \{1, 2, \ldots, n\}$.

We define $|x|_\infty := \max_{k\in\mathbb{N}} |x(k)|$ for each sequence x, where $\mathbb{N}$ denotes the set of non-negative integers, and we define $\mathcal{L}_\infty$ as the set of sequences x satisfying $|x|_\infty < \infty$.

Given $M \in \mathbb{R}^{n\times p}$ and $m \in \mathbb{R}^n$, we define $\mathrm{Set}(M, m) := \{x \in \mathbb{R}^p : Mx \preceq m\}$, which is nothing but an intersection of halfspaces in $\mathbb{R}^p$. Moreover, we define

$$\begin{aligned} &\mathrm{Proj}_{(n_1,n_2)}(M, m) := \{(H, h) \in \mathbb{R}^{\bar{n}\times((n_2-n_1+1))} \times \mathbb{R}^{\bar{n}} : \\ &\forall x \in \mathbb{R}^{n_2-n_1+1}\big(x \in \mathrm{Set}(H, h) \iff \exists v_1 \in \mathbb{R}^{n_1-1}, \\ &\exists v_2 \in \mathbb{R}^{p-n_2} : \ (v_1, x, v_2) \in \mathrm{Set}(M, m)\big)\}, \end{aligned} \tag{2}$$

where $\overline{n} \in \mathbb{N}$ and $n_1, n_2 \in \{1, 2, \ldots, p\}$ with $n_2 \geq n_1$. For each $(H, h) \in \mathrm{Proj}(M, m)$, $\mathrm{Set}(H, h)$ is the projection of $\mathrm{Set}(M, n)$ onto the coordinates from n_1 up to n_2, and it is computed using the *Fourier-Motzkin Elimination* method, c.f. Dantzig and Eaves (1973). Given a set $X \subset \mathcal{X}$, the operator co : $\mathcal{X} \rightrightarrows \mathbb{R}^{n\times p} \times \mathbb{R}^n$ is such that, for each $(H,h) \in \mathrm{co}(X)$, $\mathrm{Set}(H,h)$ is the convex hull of X provided that it has a representation as a finite intersection of halfspaces. Furthermore, we define

$$\mathfrak{d}^n(M) := \begin{cases} M & \text{if } n=1 \\ \begin{bmatrix} M & 0 \\ 0 & \mathfrak{d}^{n-1}(M) \end{bmatrix} & \text{otherwise} \end{cases} \tag{3}$$

for each matrix M and for each $n \in \mathbb{N}$ with $n \geq 1$. Similarly, we define

$$\mathfrak{s}^n(m) := \begin{cases} m & \text{if } n=1 \\ \begin{bmatrix} m \\ \mathfrak{s}^{n-1}(m) \end{bmatrix} & \text{otherwise} \end{cases} \tag{4}$$

for each vector m and for each $n \in \mathbb{N}$ with $n \geq 1$.

3. SET-VALUED OBSERVERS

In FITBOX, a Set-Valued Observer (SVO) is a class comprised of the immutable set of data (n_x, n_y, n_u, n_w) and a set of matrices $(H(k), h(k))$ such that $(x(k), w(k)) \in \mathrm{Set}(H(k), h(k))$. In this class we have implemented the projection, update and constructor methods. The projection method is an implementation of (2) that allows, in particular, the computation of matrices $(\widetilde{H}_x(k), \widetilde{h}_x(k))$ and $(\widetilde{H}_w(k), \widetilde{h}_w(k))$ from $(H(k), h(k))$ such that $x(k) \in \mathrm{Set}(\widetilde{H}_x(k), \widetilde{h}_x(k))$ and $w(k) \in (\widetilde{H}_w(k), \widetilde{h}_w(k))$. The update method makes use of the input data $u(k)$, measurements $y(k+1)$ and plant data in order to compute $(H(k+1), h(k+1))$. The implementation of this method is explained in the remainder of this section. Since the final implementation is quite intricate, we describe it with an increasing degree of complexity, as the assumptions on the plant model are sequentially dropped.

Let us assume that $\phi \in \mathcal{L}_\infty$ is such that

$$\begin{aligned} A(\phi(k)) &= A_0(k) + \sum_{i=1}^{\Delta_A} \delta_i^A(k) A_i, \\ B(\phi(k)) &= B_0(k) + \sum_{i=1}^{\Delta_B} \delta_i^B(k) B_i, \\ L(\phi(k)) &= L_0(k) + \sum_{i=1}^{\Delta_L} \delta_i^L(k) L_i, \\ C(\phi(k)) &= C_0(k) + \sum_{i=1}^{\Delta_C} \delta_i^C(k) C_i, \\ D(\phi(k)) &= D_0(k) + \sum_{i=1}^{\Delta_D} \delta_i^D(k) D_i, \\ N(\phi(k)) &= N_0(k) + \sum_{i=1}^{\Delta_N} \delta_i^N(k) N_i, \end{aligned} \tag{5}$$

for each $k \in \mathbb{N}$,for some $\{A_1, \ldots, A_{\Delta_A}\} \subset \mathbb{R}^{n_x \times n_x}$, $A_0 : \mathbb{N} \to \mathbb{R}^{n_x\times n_x}$, $\Delta_A \in \mathbb{N}$, $\delta^A : \mathbb{N} \to \mathbb{R}^{\Delta_A}$ satisfying $-1 \preceq \delta^A(k) \preceq 1$ for each $k \in \mathbb{N}$, $\{B_1, \ldots, B_{\Delta_B}\} \subset \mathbb{R}^{n_x\times n_u}$, $B_0 : \mathbb{N} \to \mathbb{R}^{n_x\times n_u}$, $\Delta_B \in \mathbb{N}$, $\delta^B : \mathbb{N} \to \mathbb{R}^{\Delta_B}$ satisfying $-1 \preceq \delta^B(k) \preceq 1$ for each $k \in \mathbb{N}$, $\{L_1, \ldots, L_{\Delta_L}\} \subset \mathbb{R}^{n_x\times n_w}$, $L_0 : \mathbb{N} \to \mathbb{R}^{n_x\times n_w}$, $\Delta_L \in \mathbb{N}$, $\delta^L : \mathbb{N} \to \mathbb{R}^{\Delta_L}$ satisfying $-1 \preceq \delta^L(k) \preceq 1$ for each $k \in \mathbb{N}$, $\{C_1, \ldots, C_{\Delta_C}\} \subset \mathbb{R}^{n_y\times n_x}$, $C_0 : \mathbb{N} \to \mathbb{R}^{n_y\times n_x}$, $\Delta_C \in \mathbb{N}$, $\delta^C : \mathbb{N} \to \mathbb{R}^{\Delta_C}$ satisfying $-1 \preceq \delta^C(k) \preceq 1$ for each $k \in \mathbb{N}$, $\{D_1, \ldots, D_{\Delta_D}\} \subset \mathbb{R}^{n_y\times n_u}$, $D_0 : \mathbb{N} \to \mathbb{R}^{n_y\times n_u}$, $\Delta_D \in \mathbb{N}$, $\delta^D : \mathbb{N} \to \mathbb{R}^{\Delta_D}$ satisfying $-1 \preceq \delta^D(k) \preceq 1$ for each $k \in \mathbb{N}$, $\{N_1, \ldots, N_{\Delta_N}\} \subset \mathbb{R}^{n_y\times n_w}$, $N_0 : \mathbb{N} \to \mathbb{R}^{n_y\times n_w}$, $\Delta_N \in \mathbb{N}$, $\delta^N : \mathbb{N} \to \mathbb{R}^{\Delta_D}$ satisfying $-1 \preceq \delta^N(k) \preceq 1$ for each $k \in \mathbb{N}$.

Notice that, if the uncertainty vectors $\delta^{A,B,L,C,D,N}$ are zero, then the evolution of the plant matrices is determined by $(A_0, B_0, L_0, C_0, D_0, N_0)$, which represents the nominal system, and the set-valued observer can be cast naturally as a set of linear inequalities following a similar approach to that of Shamma and Tu (1999). If not, then we consider the parametric uncertainty – represented by $\delta^{A,B,L,C,D,N}$ – as an additional input disturbance. In this way, the set-valued observers are still implemented by means of a set of linear matrix inequalities, using some conservative approximations, as explained in the sequel.

Let us consider that the disturbance input is $\widetilde{w}(k) \in \mathbb{R}^{\widetilde{n}_w}$ for some $\widetilde{n}_w$ and that $w(k)$ in (1), is a part of $\widetilde{w}(k)$. Furthermore, consider that $w \in \mathcal{L}_\infty$ has known bounds $w^+, w^- \in \mathcal{L}_\infty$ such that

$$w^-(k) \preceq w(k) \preceq w^+(k) \tag{6}$$

for each $k \in \mathbb{N}$, and that x is a sequence satisfying (9) and $x^-(0) \preceq x(0) \preceq x^+(0)$ for some bounded $x^+(0), x^-(0) \in \mathbb{R}^{n_x}$. Also, assume that $x(k)$ and the disturbance input $\widetilde{w}(k)$ belong to $\mathrm{Set}(H(k), h(k))$ for each $k \in \mathbb{N}$, where $(H(k), h(k)) \in \mathrm{Proj}_{(1, n_x+\widetilde{n}_w)}(\widetilde{H}(k), \widetilde{h}(k))$ with

$$\begin{aligned} \widetilde{H}(k) &:= \begin{bmatrix} I_{n_x} & 0 & -A_0(k-1) & -\widetilde{L}(k-1) \\ -I_{n_x} & 0 & A_0(k-1) & \widetilde{L}(k-1) \\ C_0(k) & \widetilde{N}(k) & 0 & 0 \\ -C_0(k) & -\widetilde{N}(k) & 0 & 0 \\ 0 & M_w(k) & 0 & 0 \\ 0 & 0 & \multicolumn{2}{c}{H(k-1)} \end{bmatrix} \\ \widetilde{h}(k) &:= \begin{bmatrix} B_0(k-1)u(k-1) \\ -B_0(k-1)u(k-1) \\ y(k) - D_0(k)u(k) \\ -y(k) + D_0(k)u(k) \\ m_w(k) \\ h(k-1) \end{bmatrix}, \end{aligned} \tag{7}$$

for each $k \geq 1$ and for some $(M_w(k), m_w(k))$ with

$$H(0) := \begin{bmatrix} I_{n_x} & 0 \\ -I_{n_x} & 0 \\ C_0(0) & \widetilde{N}(0) \\ -C_0(0) & -\widetilde{N}(0) \\ 0 & M_w(0) \end{bmatrix}, h(0) = \begin{bmatrix} x^+(0) \\ -x^-(0) \\ y(0) - D_0(0)u(0) \\ -y(0) + D_0(0)u(0) \\ m_w(0) \end{bmatrix}. \tag{8}$$

The augmented parameters $\widetilde{w}(k) \in \mathbb{R}^{\widetilde{n}_w}$, $\widetilde{L}(k)$ and $\widetilde{N}(k)$ provide additional flexibility in the description of the SVO strategy, since their values depend on the particular assumptions made during the SVO design.

It follows from (Shamma and Tu, 1999, Theorem 3.1) that $(x(k), w(k)) \in \mathrm{Set}(H(k), h(k))$ for each $k \in \mathbb{N}$ for a sequence of state values satisfying (1) and a sequence of disturbances satisfying (6).

3.1 No Parametric Uncertainty

If we consider that the system is not subject to parametric uncertainty, i.e., $\delta^{A,B,L,C,D,N} = 0$, then it follows from (1) and (5) that

$$\begin{aligned} x(k+1) &= A_0(k)x(k) + B_0(k)u(k) + L_0(k)w(k) \\ y(k) &= C_0(k)x(k) + D_0(k)u(k) + N_0(k)w(k). \end{aligned} \tag{9}$$

Then, the nominal SVO is given by (7) with $\widetilde{L} = L_0(k)$, $\widetilde{N}(k) = N_0(k)$ and

$$M_w(k) := \begin{bmatrix} I_{n_w} \\ -I_{n_w} \end{bmatrix} \quad m_w(k) := \begin{bmatrix} w^+(k) \\ w^-(k) \end{bmatrix}. \tag{10}$$

3.2 Uncertainty in the Plant Dynamics

Considering that *only* the matrix $A(k)$ is subject to parametric uncertainty, it follows from (1) and (5) that

$$\begin{aligned} x(k+1) &= \left(A_0(k) + \sum_{i=1}^{\Delta_A} \delta_i^A(k) A_i \right) x(k) + B_0(k)u(k) \\ &\quad + L_0(k)w(k) \\ &= A_0(k)x(k) + \sum_{i=1}^{\Delta_A} \delta_i^A(k) A_i x(k) + B_0(k)u(k) \\ &\quad + L_0(k)w(k). \end{aligned} \tag{11}$$

Defining $A_\Delta := [A_1 \ A_2 \ \dots \ A_{\Delta_A}]$, it follows from (11) that

$$x(k+1) = A_0(k)x(k) + A_\Delta \delta_x^A(k) + B_0(k)u(k) + L_0(k)w(k), \tag{12}$$

where $\delta_x^A(k) := \left[\delta_1^A x(k)^\top \ \dots \ \delta_{\Delta_A}^A x(k)^\top\right]^\top$.

By inspection, we may consider $\delta_x^A(k)$ as a new (augmented) disturbance, i.e., $\widetilde{w}(k) := \left[w(k)^\top \ \delta_x^A(k)^\top\right]^\top$ for each $k \in \mathbb{N}$. If $(x(k), \widetilde{w}(k))$ belongs to $\mathrm{Set}(H(k), h(k))$, then $x(k)$ belongs to $\mathrm{Set}(\widetilde{H}_x(k), \widetilde{h}_x)$ where $(\widetilde{H}_x(k), \widetilde{h}_x) \in \mathrm{Proj}_{(1,n_x)}(H(k), h(k))$. Consequently, We have that $\delta_i^A x(k) \in \mathrm{Set}(H_x(k), h_x(k))$ for each $i \in \{1, 2, \dots, \Delta_A\}$, where

$$\begin{aligned} (H_x(k), h_x(k)) \in \mathrm{co}\Big(&\mathrm{Set}(\widetilde{H}_x(k), \widetilde{h}_x(k)) \cup \\ &\mathrm{Set}(-\widetilde{H}_x(k), \widetilde{h}_x(k)) \Big), \end{aligned} \tag{13}$$

which follows from the fact that $|\delta_i^A| \leq 1$. Then, an SVO robust to parametric uncertainty in the matrix $A(k)$ is given by (7), with $\widetilde{L}(k) = \widetilde{L}^A(k)$, $\widetilde{N}(k) = \widetilde{N}^A(k)$, $M_w(k) = M_w^A(k)$ and $m_w(k) = m_w^A(k)$ where

$$\begin{aligned} &\widetilde{L}^A(k) := [L_0(k) \ A_\Delta] \qquad \widetilde{N}^A(k) := [N_0(k) \ 0] \\ &M_w^A(k) := \begin{bmatrix} I_{n_w} & 0 \\ -I_{n_w} & 0 \\ 0 & \mathfrak{d}^{\Delta_A}(H_x(k)) \end{bmatrix} \quad m_w^A(k) := \begin{bmatrix} w^+(k) \\ w^-(k) \\ \mathfrak{s}^{\Delta_A}(h_x(k)) \end{bmatrix} \end{aligned} \tag{14}$$

for each $k \in \mathbb{N}$.

3.3 Uncertainty in the Input Matrix B

If, in addition to the uncertainty in the plant dynamics, we consider that the system is subject to uncertainty in the input matrix B, then it follows from (1) and (12) that

$$\begin{aligned} x(k+1) = &A_0(k)x(k) + A_\Delta \delta_x^A(k) + B_0(k)u(k) \\ &+ B_\Delta(k)\delta^B(k) + L_0(k)w(k), \end{aligned} \tag{15}$$

where $B_\Delta(k) := [B_1 u(k) \ \dots \ B_{\Delta_B} u(k)]$. Hence, the parametric uncertainty embodied by the vector δ^B can be regarded as a disturbance. Thus, with a slight abuse of notation, let $\widetilde{w}(k) := \left[w(k)^\top \ \delta_x^A(k)^\top \ \delta^B(k)^\top\right]^\top$. Then, An SVO which is robust to parametric uncertainty in the plant dynamics and in the input matrix B is given by (7) with $\widetilde{L}(k) = \widetilde{L}^B(k)$, $\widetilde{N}(k) = \widetilde{N}^B(k)$, $M_w(k) = M_w^B(k)$ and $m_w(k) = m_w^B(k)$ where

$$\begin{aligned} &\widetilde{L}^B(k) := \left[\widetilde{L}^A(k) \ B_\Delta(k)\right] \qquad \widetilde{N}(k) := \left[\widetilde{N}^A(k) \ 0\right] \\ &M_w^B(k) := \begin{bmatrix} M_w^A(k) & 0 \\ 0 & I_{\Delta_B} \\ 0 & -I_{\Delta_B} \end{bmatrix} \quad m_w^B(k) := \begin{bmatrix} m_w^A(k) \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} \end{aligned} \tag{16}$$

for each $k \in \mathbb{N}$.

3.4 Uncertainty in the Disturbance Matrix L

In the same lines of the step-wise SVO design that we have been following thus far, we consider that the system (1) is subject not only to parametric uncertainty in matrices A and B, but also in matrix L. Then, it follows from (1), (5) and (15) that

$$\begin{aligned} x(k+1) &= A_0(k)x(k) + A_\Delta \delta_x^A(k) + B_0(k)u(k) \\ &\quad + B_\Delta(k)\delta^B(k) + L_0(k)w(k) + \sum_{i=1}^{\Delta_L} \delta_i^L(k) L_i w(k) \\ &= A_0(k)x(k) + A_\Delta \delta_x^A(k) + B_0(k)u(k) \\ &\quad + B_\Delta(k)\delta^B(k) + L_0(k)w(k) + L^\Delta \delta_w^L(k), \end{aligned} \tag{17}$$

where $L^\Delta := [L_1 \ \dots \ L_{\Delta_L}]$ and

$$\delta_w^L(k) := \left[\delta_1^L w(k)^\top \ \dots \ \delta_{\Delta_L}^L w(k)^\top\right]^\top.$$

Notice that $w(k) \in \mathrm{Set}(\widetilde{H}_w(k), \widetilde{h}_w(k))$ with

$$\widetilde{H}_w(k) := \begin{bmatrix} I_{n_w} \\ -I_{n_w} \end{bmatrix} \quad \widetilde{h}_w(k) := \begin{bmatrix} w^+(k) \\ -w^-(k) \end{bmatrix}, \tag{18}$$

for each $k \in \mathbb{N}$. Consequently, it follows that $\delta_i^L(k)w(k) \in \mathrm{Set}(H_w(k), h_w(k))$ for each $i \in \{1, 2, \dots, \Delta_L\}$ where

$$\begin{aligned} (H_w(k), h_w(k)) \in \mathrm{co}\Big(&\mathrm{Set}(\widetilde{H}_w(k), \widetilde{h}_w(k)) \cup \\ &\mathrm{Set}(-\widetilde{H}_w(k), \widetilde{h}_w(k)) \Big), \end{aligned} \tag{19}$$

for each $k \in \mathbb{N}$. Considering δ_w^L as a new disturbance input, we have that $\widetilde{w}(k) = \left[w(k)^\top \ \delta_x^A(k) \ \delta^B(k) \ \delta_w^L(k)\right]$ for each $k \in \mathbb{N}$. Then, an SVO for the system (1) that is robust to parametric uncertainty in the matrices A, B, and L is given by (7) with $\widetilde{L}(k) = \widetilde{L}^L(k)$, $\widetilde{N}(k) = \widetilde{N}^L(k)$, $M_w(k) = M_w^L(k)$ and $m_w(k) = m_w^L(k)$ where

$$\begin{aligned} &\widetilde{L}^L(k) := \left[\widetilde{L}^B(k) \ L_\Delta\right] \qquad \widetilde{N}^L(k) := \left[\widetilde{N}^B(k) \ 0\right] \\ &M_w^L(k) := \begin{bmatrix} M_w^B(k) & 0 \\ 0 & \mathfrak{d}^{\Delta_L}(H_w(k)) \end{bmatrix} \quad m_w^L(k) := \begin{bmatrix} m_w^B(k) \\ \mathfrak{s}^{\Delta_L}(h_w(k)) \end{bmatrix} \end{aligned} \tag{20}$$

for each $k \in \mathbb{N}$.

3.5 Uncertainty in the Output Matrix C

Assuming that the system in (1) is subject to parametric uncertainty in the matrices A, B, L and C, it follows from (5) and (15) that

$$
\begin{aligned}
y(k) = &\left(C_0(k) + \sum_{i=1}^{\Delta_C} \delta_i^C(k) C_i\right) x(k) + D_0(k)u(k) \\
&+ N_0(k)w(k) \\
=&C_0(k)x(k) + \left(\sum_{i=1}^{\Delta_C} \delta_i^C(k) C_i\right) \big(A_0(k-1)x(k-1) \\
&+ A_\Delta \delta_x^A(k-1) + B_0(k-1)u(k-1) \\
&+ B_\Delta(k-1)\delta^B(k-1) + L_0(k-1)w(k-1) \\
&+ L^\Delta \delta_w^L(k-1)\big) + D_0(k)u(k) + N_0(k)w(k).
\end{aligned} \tag{21}
$$

It follows from (21) that $\delta_i^C(k)x(k-1)$, $\delta_i^C(k)\delta_x^A(k-1)$, $\delta_i^C(k)\delta^B(k-1)$, $\delta_i^C(k)w(k-1)$ and $\delta_i^C(k)\delta_w^L(k-1)$ for each $i \in \{1, 2, \dots, \Delta_C\}$ can be considered as new input disturbances. Notice that, for each $i \in \{1, 2, \dots, \Delta_C\}$, we have that $\delta_i^C(k)x(k-1) \in \mathrm{Set}(H_x(k-1), h_x(k-1))$ with $\mathrm{Set}(H_x(k), h_x(k))$ given by (13) and $\delta_i^C(k)w(k-1) \in \mathrm{Set}(H_w(k-1), h_w(k-1))$ with $(H_w(k-1), h_w(k-1))$ given by (19). Moreover, since $\left|\delta_i^C(k)\delta_j^A(k-1)\right| \le 1$ for each $i \in \{1, 2, \dots, \Delta_C\}$, $j \in \{1, 2, \dots, \Delta_A\}$ and $\left|\delta_i^C(k)\delta_j^L(k-1)\right| \le 1$ for each $i \in \{1, 2, \dots, \Delta_C\}$, $j \in \{1, 2, \dots, \Delta_L\}$, it follows that $\delta_i^C(k)\delta_j^A(k-1)x(k-1) \in \mathrm{Set}(H_x(k-1), h_x(k-1))$ for each $i \in \{1, 2, \dots, \Delta_C\}$, $j \in \{1, 2, \dots, \Delta_A\}$ and $\delta_i^C(k)\delta_j^L(k-1)w(k-1) \in \mathrm{Set}(H_w(k-1), h_w(k-1))$ for each $i \in \{1, 2, \dots, \Delta_C\}$, $j \in \{1, 2, \dots, \Delta_L\}$. It can be concluded from these considerations that, At the cost of added conservatism, an SVO for the system in (1), which is robust to parametric uncertainty in the matrices A, B, L and C, is given by (7) with $\widetilde{L}(k) = \widetilde{L}^C(k)$, $\widetilde{N}(k) = \widetilde{N}^C(k)$, $M_w(k) = M_w^C(k)$ and $m_w(k) = m_w^C(k)$ where

$$
\begin{aligned}
&\widetilde{L}^C(k) := \left[\widetilde{L}^L(k)\ 0\ 0\ 0\ 0\ 0\ 0\right] \\
&\widetilde{N}^C(k) := \left[\widetilde{N}^L(k)\ C_\Delta\, \mathfrak{d}^{\Delta_C}(A_0(k-1))\ C_\Delta\, \mathfrak{d}^{\Delta_C}(A_\Delta)\ \dots\right. \\
&\dots\ C_\Delta\, \mathfrak{d}^{\Delta_C}(B_0(k-1)u(k-1))\ C_\Delta\, \mathfrak{d}^{\Delta_C}(B_\Delta(k-1))\ \dots \\
&\left.\dots\ C_\Delta\, \mathfrak{d}^{\Delta_C}(L_0(k-1))\ C_\Delta\, \mathfrak{d}^{\Delta_C}(L_\Delta)\right] \\
&M_w^C(k) := \begin{bmatrix} M_w^L(k) & 0 & 0 & 0 & 0 \\ 0 & \mathfrak{d}^{\Delta_C(1+\Delta_A)}(H_x(k-1)) & 0 & 0 & 0 \\ 0 & 0 & I_{\Delta_C} & 0 & 0 \\ 0 & 0 & -I_{\Delta_C} & 0 & 0 \\ 0 & 0 & 0 & \mathfrak{d}^{\Delta_C}\left(\begin{bmatrix} I_{\Delta_B} \\ -I_{\Delta_B}\end{bmatrix}\right) & 0 \\ 0 & 0 & 0 & 0 & \mathfrak{d}^{\Delta_C(1+\Delta_L)}(H_w(k)) \end{bmatrix} \\
&m_w^C(k) := \begin{bmatrix} m_w^L(k) \\ \mathfrak{s}^{\Delta_C(1+\Delta_A)}(h_x(k-1)) \\ \mathbf{1} \\ \mathfrak{s}^{\Delta_C(1+\Delta_A)}(h_w(k-1)) \end{bmatrix}
\end{aligned} \tag{22}
$$

for each $k \ge 1$, where $C_\Delta := [C_1\ \dots\ C_{\Delta_C}]$.

A clear drawback of this approach is that disturbances in the output matrix C increase significantly the complexity of the associated SVO and should be avoided whenever possible. An alternative solution consists in increasing the bounds of the noise levels associated to a given sensor.

3.6 Uncertainty in the Input Matrix D

The case where D in (1) is subject to parametric uncertainty is fairly similar to the case where B is subject to parametric uncertainty. In this case, the vector δ^D can be taken directly as a new disturbance input, subject to the constraint $-1 \preceq \delta^D(k) \preceq 1$ for each $k \in \mathbb{N}$. Then, an SVO that is robust to parametric uncertainty on the matrices A, B, C, L, D is given by (7), with $\widetilde{L}(k) = \widetilde{L}^D(k)$, $\widetilde{N}(k) = \widetilde{N}^D(k)$, $M_w(k) = M_w^D(k)$ and $m_w(k) = m_w^D(k)$ where

$$
\begin{aligned}
&\widetilde{L}^D(k) := \left[\widetilde{L}^C(k)\ 0\right] \qquad \widetilde{N}^D(k) := \left[\widetilde{N}^C(k)\ D_\Delta\right] \\
&M_w^D(k) := \begin{bmatrix} M_w^C(k) & 0 \\ 0 & I_{\Delta_D} \\ 0 & -I_{\Delta_D} \end{bmatrix} \quad m_w^D(k) := \begin{bmatrix} m_w^C(k) \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}
\end{aligned} \tag{23}
$$

for each $k \in \mathbb{N}$, with $D_\Delta := [D_1\ \dots\ D_{\Delta_D}]$.

3.7 Parametric Uncertainty in the Noise Matrix N

The case where the matrix N is subject to parametric uncertainty is handled very similarly to the case where L is subject to parametric uncertainty, because $\delta_i^D(k)w(k) \in \mathrm{Set}(H_w(k), h_w(k))$ for each $k \in \mathbb{N}$. Then, an SVO which is robust to parametric uncertainty on the matrices A, B, C, L, D and N is given by (7), with

$$
\begin{aligned}
&\widetilde{L}(k) := \left[\widetilde{L}^D(k)\ 0\right] \qquad \widetilde{N}(k) := \left[\widetilde{N}^D(k)\ N_\Delta\right] \\
&M_w(k) := \begin{bmatrix} M_w^D(k) & 0 \\ 0 & \mathfrak{d}^{\Delta_N}(H_w(k)) \end{bmatrix} \quad m_w(k) := \begin{bmatrix} m_w^D(k) \\ \mathfrak{s}^{\Delta_N}(h_w(k)) \end{bmatrix}
\end{aligned} \tag{24}
$$

for each $k \in \mathbb{N}$, with $N_\Delta := [N_1\ \dots\ N_{\Delta_N}]$. This concludes the design of SVOs that are robust to parametric uncertainty.

After tedious but straightforward manipulations of (24), the SVOs can be designed to take into account constraints on the state, disturbances, and measurements, from previous time instants, which can significantly reduce the conservatism of the estimates. For further details on this topic, the reader is referred to Rosa (2011). In particular, a so-called *horizon* K is used. In other words, to determine $\mathrm{Set}(H(k), h(k))$ we make use not only of $(H(k-1), h(k-1))$ as suggested by (7), but also $(H(k-2), h(k-2))$, $(H(k-3), h(k-3))$, $\dots$, $(H(k-K), h(k-K))$.

3.8 Wind Turbine Example

In this section, we make use of the blade subsystem of a wind turbine to illustrate the functions implemented in the SVO class shipped with FITBOX. The system model is taken from Odgaard et al. (2013) and its continuous-time description is given by

$$
\begin{aligned}
\dot{x} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} x + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} u - \frac{1}{2}\begin{bmatrix} 0 & 0 \\ \omega_n^2 & \omega_n^2 \end{bmatrix} w \\
y &= \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w,
\end{aligned} \tag{25}
$$

where $\omega_n = 11.11$ deg/s, $\xi = 0.6$, $x(t) \in \mathbb{R}^2$ denotes the angular position/velocity of the blade pitch, $u(t) \in$

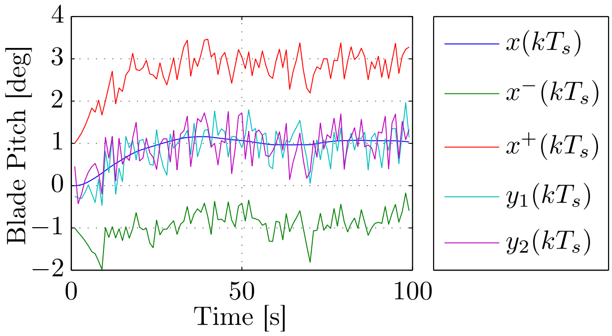

Fig. 1. Simulation results of the system in (26).

$\mathbb{R}$ denotes the reference pitch and $w(t) \in \mathbb{R}^2$ denotes the input noise, which is Gaussian white noise with a standard deviation of 0.4 deg. The noise affects both the output and the dynamics of the system because it is assumed that there exists, by default, an inner–loop controller that feeds noise back into the system. We find its corresponding discrete-time description using a *sample–and–hold* strategy with a sample period of $T_s = 0.01$ s, which yields a system of the form (9) with

$$\begin{aligned} A(k) &\approx \begin{bmatrix} 0.9941 & 0.0093 \\ -1.1532 & 0.8695 \end{bmatrix} & B(k) &\approx \begin{bmatrix} 0.0059 \\ 1.1532 \end{bmatrix} \\ L(k) &\approx \begin{bmatrix} -0.0030 & -0.0030 \\ -0.5766 & -0.5766 \end{bmatrix} & C(k) &= \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \\ D(k) &= 0 & N(k) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{26}$$

In the SVO implementation provided in Casau et al. (2014a) the user may select the following options:

Approximation none/hypercube(default);
Solver MATLAB (default)/CPLEX;
Angular Tolerance A positive scalar $angTol > 0$ ($angTol = 10^{-2}$ by default);
Numeric Tolerance A positive scalar $numTol > 0$ ($numTol = 10^{-6}$ by default);
Accelerator on/off (default).

In Casau et al. (2014a), we provide full functionality for the hypercube approximation. This approximation vastly decreases runtime operation as both Fourier-Motzkin Elimination and Convex Hull operations are significantly simplified when this method is used.

For simple dynamic systems, such as the one described in this example, the linear solver shipped with MATLAB is able to provide set-valued state estimates using a reasonably low numeric tolerance. However, for more complex systems, we recommend that the CPLEX solver IBM (2014) is used instead. Finally, the *accelerator* option provides a tweak to the implementation that avoids the Fourier-Motzkin Elimination at each time step. Also, the horizon K is set to 1, by default.

In Figure 1, we present the results of a simulation run for the system (26) with a time span of 1 s, $u(k) = 1$ for each $k \in \mathbb{N}$ and zero initial conditions. We have considered constant bounds on the input noise with magnitude of 5σ and $x^{\pm}(0) = \pm 1$, with $K = 1$.

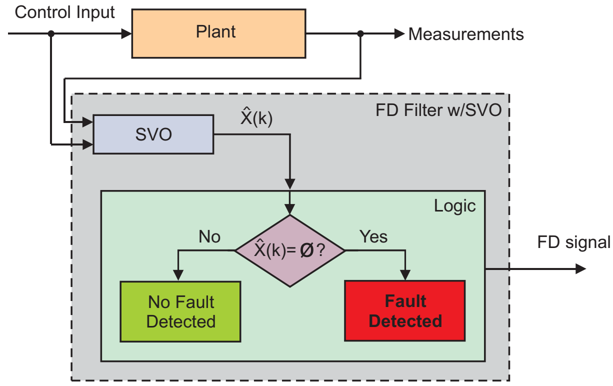

Fig. 2. Fault detection using SVOs.

4. FDI LOGIC – MODEL FALSIFICATION

The Fault Detection and Isolation (FDI) strategy that is implemented in FITBOX follows a *model falsification* approach that relies on the set–valued state estimates of the SVOs introduced in Section 3. Given a set of input/output data y, u, respectively, if a particular SVO model matches the plant model, then the corresponding SVO returns a non-empty set-valued state estimate. If there is a mismatch between the models, then the corresponding SVO *may* return the empty set as the set-valued state estimate, indicating that there is no state $x(k)$ for the current time k that satisfies (1) for the given data. This is illustrated in Figure 2.[1] This technique has been widely used in our previous studies, as documented in Casau et al. (2014b), Silvestre et al. (2013), Rosa et al. (2012). Moreover, we need to make a clear distinction between the two very different classes of SVOs presented next.

Definition 1. An SVO is said to be of *class* I if it is compatible with the nominal operation of the plant. It is said to be of *class* II otherwise. □

The Fault Isolation Module implemented in FITBOX is comprised of two main parts: a bank of SVOs, and an update function, which implements the FDI logic. In particular, it requires the definition of a minimum of three distinct SVOs:

Nominal: Class I SVO which, ideally, is compatible with and only with the nominal operation of the plant;
Global: Class I SVO which is compatible with every plant model;
Faulty: An SVO that is compatible with the operation of the plant under the influence of a given fault. This SVO can be either a class I or a class II SVO, depending on the nature of the fault it describes. If the fault model includes the nominal model as a particular case, then it should be a class I SVO, otherwise it should be a class II SVO.

[1] Whether the SVO returns the empty set as the set-valued state estimate or not, depends on the distinguishability properties between the plant model and the SVO model. In FITBOX, there is no automated feature to check distinguishability, thus it is not discussed in the paper. The reader is referred to Rosa and Silvestre (2011) for more information on this topic.

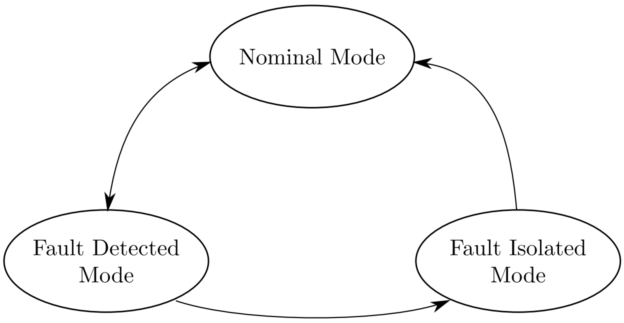

Fig. 3. Modes of operation of the FDI logic implemented in FITBOX.

Beyond the first three SVOs, there may be any number of SVOs in the bank, depending on the number of faults under consideration. Below, we present the pseudo–code that governs the operation of the fault isolation module where S denotes the bank of SVOs indexed by a unique tag (in particular, we make use of the tags *Nominal* and *Global* for the nominal and global SVOs, respectively), $T_r > 0$ denotes the timeout limit, active is a logical array that stores the operation state of each SVO, mode is a variable that determines the operation mode of the module (Nominal, Fault Detected, Fault Isolated) and fault denotes the tag of the active SVO. The method `Class(S)` retrieves the class of the SVO **S**, `XData(S)` retrieves $x^{\pm}(k)$ from **S** and `Empty(S)` determines whether the set-valued state estimate of **S** is empty.

The execution of Algorithm 1 is fairly straightforward: if the nominal SVO provides the empty set-valued state estimate then a fault is detected and a timer is initialized. The system moves to the Fault Detected mode if more than 2 SVOs are active (Algorithm 3), or to the Fault Isolated mode otherwise (Algorithm 4). From the Fault Detected mode, the execution either moves to the Nominal mode if a timeout T_r is reached or to the Fault Isolated mode if only 2 SVOs are active: the Global SVO (which is always active) and the fault specific SVO. From the Fault Isolated mode, the system comes back to the Nominal mode if a timeout is reached or the Global SVO remains as the only active SVO. When entering the nominal mode, the class I SVOs are set to active and the class II SVOs are set to inactive. A simple representation of possible mode transitions is provided in Figure 3.

5. WIND TURBINE NUMERICAL EXAMPLE

In this section, we describe the application of FITBOX to the FDI of the pitch subsystem of a wind turbine that was presented in Section 3.8. The code used in this example may be found in Casau et al. (2014a).

The set of faults that we consider is taken from Odgaard et al. (2013) and these are given next.

5.1 Fault #1: Fixed Output

Let us consider that y_1 is fixed at a given output value y_F. We can design a class II SVO for the system (25) subject to this fault using the following continuous-time model:

Input: A_0, B_0, L_0, C_0, D_0, N_0; – nominal plant data; $w^{\pm}(k)$– disturbance bounds, u, y– input/output data; S – a bank of SVOs.
Output: fault– a sequence in S that stores the active SVO as a function of time.

```
for each SVO S in S satisfying Class(S) = I do
    S ← (x⁻(0), x⁺(0));
    active[S] ← true;
end
mode ← Nominal;
for each k ∈ ℕ do
    for each SVO S in S satisfying active[S] = true do
        S ← Update(A0(k), B0(k), L0(k), C0(k), D0(k),
            N0(k), w⁺(k), w⁻(k), u(k), y(k));
        if Empty(S) then
            active[S] ← false;
        end
    end
    if active[Global] = false then
        return Error;
    end
    if mode = Nominal then
        // Nominal Mode Routine
        ⋮
    end
    if mode = Fault Detected then
        // Fault Detected Mode Routine
        ⋮
    end
    if mode = Fault Isolated then
        // Fault Isolated Mode Routine
        ⋮
    end
end
```

Algorithm 1: Update method for the Fault Isolation Module

```
if active[Nominal] = false then
    for each SVO S in S satisfying Class(S) = II do
        S ← XData(Global);
        S ← Update(A0(k), B0(k), L0(k), C0(k), D0(k),
            N0(k), w⁺(k), w⁻(k), u(k), y(k));
        if ¬Empty(S) then
            active[S] ← true;
        end
    end
    timer ← 0;
    mode ← Fault Detected;
else
    fault[k] ← Nominal;
end
if Global is the only active SVO then
    Reset class I SVOs;
    mode ← Nominal;
end
```

Algorithm 2: Nominal Mode Routine

if *there are only 2 active SVOs: Global and* $\mathbf{S}_F$ **then**
  fault[k] $\leftarrow \mathbf{S}_F$;
  mode $\leftarrow$ Fault Isolated;
else if timer $\geq T_r$ **then**
  Reset class I SVOs;
  mode $\leftarrow$ Nominal;
else
  fault[k] $\leftarrow$ fault[$k-1$];
  timer $++$;
end

Algorithm 3: Fault Detected Mode Routine

if *only Global is active OR (*timer $\geq T_r$ *AND* `Class(fault)` $=$ *I)* **then**
  Reset class I SVOs;
  mode $\leftarrow$ Nominal;
  fault[k]$\leftarrow$ Nominal;
else
  fault[k] $\leftarrow$ fault[$k-1$];
  timer $++$;
end

Algorithm 4: Fault Isolated Mode Routine

$$\begin{aligned}\dot{x} &= \left(\begin{bmatrix}0 & 1\\ -\omega_n^2 & -2\xi\omega_n\end{bmatrix} - \frac{1}{2}\begin{bmatrix}0\\ \omega_n^2\end{bmatrix}[1\ 0]\right)x + \begin{bmatrix}0\\ \omega_n^2\end{bmatrix}u\\ &\quad - \frac{1}{2}\begin{bmatrix}0 & 0\\ \omega_n^2 & \omega_n^2\end{bmatrix}w\\ y &= \begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix}x + \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}w,\end{aligned} \tag{27}$$

with $w_1^-(k) \leq y_F \leq w_1^+(k)$ and $w_1^\pm$ denote the upper/lower bounds on the fixed output value. From (27), we see that the output y_1 does not depend on the state and that there is a change in the plant dynamics due to the feedback of erroneous measurements back into the system. In the simulations, we consider $y_F = 5$ deg.

5.2 Fault #2: Change in the Sensor Gain

Suppose that $y_2 = [k\ 0]x$ with $k \in \mathbb{R}$, instead of $y_2 = [1\ 0]x$. In this case, we design a class II SVO using the continuous-time model

$$\begin{aligned}\dot{x} &= \left(\begin{bmatrix}0 & 1\\ -\omega_n^2 & -2\xi\omega_n\end{bmatrix} - \frac{1}{2}\begin{bmatrix}0\\ \omega_n^2\end{bmatrix}[k-1\ \ 0]\right)x + \begin{bmatrix}0\\ \omega_n^2\end{bmatrix}u\\ &\quad - \frac{1}{2}\begin{bmatrix}0 & 0\\ \omega_n^2 & k\omega_n^2\end{bmatrix}w\\ y &= \begin{bmatrix}1 & 0\\ k & 0\end{bmatrix}x + \begin{bmatrix}1 & 0\\ 0 & k\end{bmatrix}w,\end{aligned} \tag{28}$$

with the same upper/lower bounds $w^\pm$ on the input w that we have considered for the nominal model. This SVO design requires that the change in the sensor gain is known precisely which is seldom the case. In order to deal with the uncertainty in the sensor gain, one may add an uncertainty matrix C_1 to the SVO. In the simulations, we consider that $k = 1.2$.

5.3 Fault #3: Progressive Change in the Plant Dynamics

Suppose that the parameters (ω_n, ξ) of the nominal plant model (25) change continuously over time to (ω_2, ξ_2). In this case, one must design an SVO that is both compatible with the nominal model and the faulty model. This is achieved by designing an SVO using the nominal model (25) with parametric uncertainty in the matrices A, B and L which are those that are affected by changes in the system dynamics. These uncertainties should be large enough to encompass the changes inflicted on the system by the given fault.

In order to test the proposed FDI strategy, we ran 100 Monte-Carlo simulations for each fault scenario, using the input

$$u(t) = 8\sin(6t) + 7,$$

that enhances the distinguishability across different system models. This active input excitation strategy is usually referred to active fault diagnosis. The results of this test are provided in Table 1. From these results it is possible to conlcude that fault #1 is detected and isolated as soon as it is triggered. The abrupt change in sensor gain is typically detected and isolated within 1 s. However, fault no. 3 is slowly developing, thus it takes more than 7 s to detect and isolate. In the simulations we consider that ω_n changes from 11.11 to 3.42 deg/s and ξ changes from 0.6 to 0.9 over 30 seconds.

Fault no.	Median Detection Time [s]	Max. Detection Time [s]	Median Isolation Time [s]	Max. Isolation Time [s]
1	0	0	0	0
2	0.31	0.45	0.41	6.67
3	7.23	8.25	7.645	8.67

Table 1. Fault detection simulations results

It should be pointed out that, even with the hypercube approximation, the proposed strategy demands vast computational resources and, at this point, it is not possible to run the proposed algorithm in real-time. In particular, with the computational resources at our disposal, for each 10 ms in simulation time, the algorithm takes roughly 110 ms. Nevertheless, for a simpler system, we have successfully implemented these algorithms in real-time, as shown in Rosa et al. (2012).

6. CONCLUSION

In this paper, we have described the implementation of the basic functionality of Set-Valued Observers (SVOs) and of the logic for Fault Detection and Isolation. This functionality is implemented in FITBOX, a MATLAB toolbox that we have made publicly available. This toolbox is distributed with two examples that demonstrate its capabilities for the fault detection and isolation of the pitch subsystem of a wind turbines. We have also provided a description of these examples in this paper.

7. ACKNOWLEDGEMENTS

We would like to thank Sérgio Brás and Daniel Silvestre for their valuable comments and suggestions during the development and implementation of the algorithms available in FITBOX.

REFERENCES

Beard, R.V. (1971). *Failure accomodation in linear systems through self-reorganization.* Thesis, Massachusetts Institute of Technology. Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics. Thesis. 1971. Ph.D.

Casau, P., Rosa, P., and Silvestre, C. (2014a). FITBOX – fault isolation toolbox. URL `https://github.com/pcasau/FITBOX`.

Casau, P., Rosa, P., Tabatabaeipour, S.M., Silvestre, C., and Stoustrup, J. (2014b). A set-valued approach to FDI and FTC of wind turbines (accepted for publication). *IEEE Transactions on Control Systems Technology*.

Chen, J. and Patton, R.J. (2012). *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Springer.

Combastel, C. and Raka, S.A. (2009). A set-membership fault detection test with guaranteed robustness to parametric uncertainties in continuous time linear dynamical systems. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 1192–1197. doi:10.3182/20090630-4-ES-2003.00195.

Dantzig, G.B. and Eaves, B.C. (1973). Fourier-motzkin elimination and its dual. *Journal of Combinatorial Theory, Series A*, 14(3), 288–297. doi:10.1016/0097-3165(73)90004-6.

De Persis, C. and Isidori, A. (2001). A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46(6), 853–865. doi:10.1109/9.928586.

Frank, P.M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica*, 26(3), 459–474. doi:10.1016/0005-1098(90)90018-D.

Frank, P.M. and Ding, X. (1994). Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. *Automatica*, 30(5), 789–804. doi:10.1016/0005-1098(94)90169-4.

Gertler, J. and Kunwer, M. (1995). Optimal residual decoupling for robust fault diagnosis. *International Journal of Control*, 61(2), 395–421. doi:10.1080/00207179508921908.

Hargreaves, G.I. (2002). Interval analysis in MATLAB. Technical report, Department of Mathematics, University of Manchester, Manchester.

IBM (2014). IBM CPLEX optimizer. URL `http://www.ibm.com/software/products/en/ibmilogcpleoptistud`.

Ingimundarson, A., Bravo, J.M., Puig, V., Alamo, T., and Guerra, P. (2009). Robust fault detection using zonotope-based set-membership consistency test. *International Journal of Adaptive Control and Signal Processing*, 23(4), 311–330. doi:10.1002/acs.1038.

Jauberthie, C., Verdière, N., and Travé-Massuyès, L. (2013). Fault detection and identification relying on set-membership identifiability. *Annual Reviews in Control*, 37(1), 129–136. doi:10.1016/j.arcontrol.2013.04.002.

Jones, H.L. (1973). *Failure detection in linear systems.* Thesis, Massachusetts Institute of Technology.

Kurzhanskiy, A. and Varaiya, P. (2006). Ellipsoidal toolbox (ET). In *Proceedings of the 45th IEEE Conference on Decision and Control*, 1498–1503. doi:10.1109/CDC.2006.377036.

Odgaard, P., Stoustrup, J., and Kinnaert, M. (2013). Fault-tolerant control of wind turbines: A benchmark model. *IEEE Transactions on Control Systems Technology*, 21(4), 1168–1182. doi:10.1109/TCST.2013.2259235.

Puig, V. (2010). Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies. *International Journal of Applied Mathematics and Computer Science*, 20(4), 619–635. doi:10.2478/v10006-010-0046-y.

Rosa, P. (2011). *Multiple-Model Adaptive Control of Uncertain LPV Systems.* Ph.D. thesis, PhD thesis, Instituto Superior Técnico, Lisbon, Portugal.

Rosa, P. and Silvestre, C. (2011). On the distinguishability of discrete linear time-invariant dynamic systems. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 3356–3361. doi:10.1109/CDC.2011.6159300.

Rosa, P., Simao, T., Lemos, J.M., and Silvestre, C. (2012). Multiple-model adaptive control of an air heating fan using set-valued observers. In *Proceedings of the 20th Mediterranean Conference on Control Automation (MED)*, 469–474. doi:10.1109/MED.2012.6265682.

Shamma, J. and Tu, K.Y. (1999). Set-valued observers and optimal disturbance rejection. *IEEE Transactions on Automatic Control*, 44(2), 253–264. doi:10.1109/9.746252.

Silvestre, D., Rosa, P., Cunha, R., Hespanha, J.P., and Silvestre, C. (2013). Gossip average consensus in a byzantine environment using stochastic set-valued observers. In *Proceedings of the IEEE 52nd Annual Conference on Decision and Control (CDC)*, 4373–4378. doi:10.1109/CDC.2013.6760562.

Varga, A. (2006). A fault detection toolbox for MATLAB. In *Proceedings of the IEEE Computer Aided Control System Design*, 3013–3018. doi:10.1109/CACSD-CCA-ISIC.2006.4777118.