Control Engineering Practice 21 (2013) 242–252

Contents lists available at SciVerse ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Fault detection and isolation of LPV systems using set-valued observers: An application to a fixed-wing aircraft

Paulo Rosa [a,]*, Carlos Silvestre [a,b]

[a] Institute for Systems and Robotics - Instituto Superior Tecnico, North Tower, 8th floor, Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal
[b] Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Taipa, Macau

ARTICLE INFO

Article history:
Received 26 September 2011
Accepted 20 October 2012
Available online 19 November 2012

Keywords:
Fault detection
Fault isolation
Fault diagnosis
Time-varying systems
Uncertain dynamic systems
Discrete-time systems

ABSTRACT

A novel fault detection and isolation (FDI) method using set-valued observers (SVOs), for uncertain linear parameter-varying (LPV) systems, is introduced. The proposed method relies on SVO-based model invalidation to discard models incompatible with measured data. When compared to the most common strategies in the literature, the suggested approach: (i) under suitable conditions, guarantees false alarms are avoided; (ii) unlike residual-based architectures, does not require the computation of thresholds to declare faults; (iii) has applicability to a wide class of plants. The performance of the proposed approach is assessed in simulation, using the full nonlinear model of a fixed-wing aircraft longitudinal dynamics.

1. Introduction

The field of fault detection and isolation (FDI) has been studied since the early 1970s Willsky (1976), and several techniques have, since then, been applied to different types of systems. An FDI device is key in several applications and, in particular, in those that are *safety critical*. Common examples of systems equipped with FDI devices include aircrafts and a wide range of industrial processes such as the ones described in the following references—Blanke, Izadi-Zamanabadi, Bogh, and Lunau (1997), Blanke, Staroswiecki, and Wu (2001), Isermann (1997), Patton and Chen (1997), Frank and Ding (1997), Esteban (2004), Collins and Tinglun (2001), Alwi and Edwards (2008), Longhi and Moteriù (2009), Mattone and De Luca (2006). For a survey of FDI methods in the literature, see, for instance, Hwang, Kim, Kim, and Eng Seah (2010). An FDI system must be able to bear with different types of faults in sensors and/or actuators, which can occur abruptly or slowly in time. Moreover, model uncertainty (such as unmodeled dynamics) and disturbances must never be interpreted as faults. Notwithstanding the hundreds (or maybe thousands!) of papers in the literature concerning this topic, there are still some open questions related to the performance guarantees provided by these devices.

An active deterministic model-based fault detection (FD) system (see Esteban, 2004, for a description of the typical FD classes available in the literature) is usually composed of two parts: a filter that generates residuals, which should be *large* under faulty environments; and a decision threshold, which is used to decide whether a fault is present or not—see Willsky (1976), Patton and Chen (1997), Esteban (2004), Frank and Ding (1994), Massoumnia (1986b), Besançon (2003), Bokor and Balas (2004), Blanke, Kinnaert, Lunze, Staroswiecki, and Schröder (2006), Puig, Quevedo, Escobet, Nejjari, and de las Heras (2008), Meskin and Khorasani (2009), Wang, Wang, and Shi (2009), Narasimhan, Vachhani, and Rengaswamy (2008), Ducard (2009) and references therein. The isolation of the fault can, in some cases, be done using a similar approach, i.e., by designing filters for families of faults, and identifying the most likely fault as that associated to the filter with smaller residuals.

The main idea in such architectures stems from the design of filters that are more sensitive to faults than to disturbances and model uncertainty. This can be achieved, for instance, by using geometric considerations regarding the plant, as in Massoumnia (1986a, 1986b), Longhi and Moteriù (2009), Bokor and Balas (2004), or by optimizing a particular norm minimization objective, such as the $\mathcal{H}_\infty$- or l_1-norm—see Edelmayer, Bokor, and Keviczky (1994), Frank and Ding (1997), Niemann and Stoustrup (2001), Marcos, Ganguli, and Balas (2005), Collins and Tinglun (2001). The latter approach provides, in general, important robustness properties, as stressed in Edelmayer et al. (1994), Mangoubi, Appleby, Verghese, and Vander Velde (1995), Patton and Chen (1997) and Esteban (2004), by explicitly accounting for model uncertainty.

The FDI strategy proposed in this paper uses a different philosophy. Instead of identifying the most likely model of the faulty plant, one discards models that are not compatible with the observations. As shown in the sequel, this method guarantees that there will not be false alarms, as long as the model of the non-faulty plant remains valid. I.e., if the assumptions regarding the bounds on the exogenous disturbances are not violated, and the model of the dynamics of the plant is valid, then no fault is declared. Moreover, one need not

* Corresponding author. Tel.: +351 218418054; fax: +351 218418291.
E-mail addresses: prosa@isr.ist.utl.pt (P. Rosa), cjs@isr.ist.utl.pt (C. Silvestre).

0967-0661/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.conengprac.2012.10.005

compute the decision threshold used to declare whether or not a fault has occurred. To this end, the technique introduced in Rosa, Silvestre, Shamma, and Athans (2009, 2010), which is based upon the use of set-valued observers (SVOs) – see Witsenhausen (1968), Schweppe (1968, 1973), Milanese and Vicino (1991) and Shamma and Tu (1999) and references therein for an overview on SVOs – is extended.

In this paper, an application example of a new FDI method based on SVOs is provided, and the performance of the aforementioned technique when applied to the detection of faults in an aircraft is addressed. The performance of the approach is assessed in simulation, by deliberately generating faults in the nonlinear aircraft model. The key criteria of this evaluation are the time required to diagnose a failure, and the robustness of the method against model uncertainty and exogenous disturbances.

The remainder of this paper is organized as follows: Section 2 introduces the robust SVOs that are going to be used for FDI; Section 3 presents the methodology to design FD filters using SVOs, and discusses some of the approaches to modeling several types of faults; Section 4 extends the ideas in Section 3 for isolating the faults; Section 5 presents the full nonlinear longitudinal aircraft dynamic model and the corresponding LPV approximation; simulation results for the nonlinear dynamic model of the longitudinal dynamics of an aircraft are presented in Section 6; and finally, Section 7 summarizes some of the conclusions regarding this paper.

2. Set-valued observers

As shown in the sequel, the problem of “disqualifying” dynamic models of a system can be tackled using set-valued observers (SVOs)—see Witsenhausen (1968), Schweppe (1968, 1973) and Milanese and Vicino (1991). One assumes that the non-faulty plant can be represented by an uncertain (possibly time-varying) discrete-time linear system, with uncertain initial conditions, and excited by bounded but unknown exogenous disturbances, i.e.,

$$\begin{cases} \boldsymbol{x}(k+1)=(\mathbf{A}(k)+\mathbf{A}_{\Delta}(k))\boldsymbol{x}(k)+\mathbf{L}_d(k)\boldsymbol{d}(k)+(\mathbf{B}(k)+\mathbf{B}_{\Delta}(k))\boldsymbol{u}(k), \\ \boldsymbol{y}(k)=(\mathbf{C}(k)+\mathbf{C}_{\Delta}(k))\boldsymbol{x}(k)+\boldsymbol{n}(k), \end{cases} \tag{1}$$

where $\boldsymbol{x}(k)\in\mathbb{R}^n$, $\boldsymbol{d}(k)\in\mathbb{R}^{n_d}$, $\boldsymbol{u}(k)\in\mathbb{R}^{n_u}$, $\boldsymbol{y}(k)\in\mathbb{R}^{n_y}$, $\boldsymbol{x}(0)=\boldsymbol{x}_o$, $\boldsymbol{x}_o\in X(0)$, $\boldsymbol{d}(k)$ with $\|\boldsymbol{d}(k)\| := \max_i|\boldsymbol{d}_i(k)|\le 1$ are the disturbances, $\boldsymbol{n}(k)$ with $\|\boldsymbol{n}(k)\| := \max_i|n_i(k)|\le\overline{n}$ is the sensors noise, $\boldsymbol{u}(k)$ is the control input, $\boldsymbol{y}(k)$ is the measured output, $\boldsymbol{x}(k)$ is the state of the system and $X(0) := \mathrm{Set}(\mathbf{M}_o,\boldsymbol{m}_o)$, where, for any positive integers ℓ_1 and ℓ_2, and for any matrix $\mathbf{M}\in\mathbb{R}^{\ell_1\times\ell_2}$ and vector $\boldsymbol{m}\in\mathbb{R}^{\ell_1}$,

$$\mathrm{Set}(\mathbf{M},\boldsymbol{m}) := \{\boldsymbol{q} : \mathbf{M}\boldsymbol{q}\le\boldsymbol{m}\} \tag{2}$$

represents a convex polytope. In $X(0)$, $\mathbf{M}_o$ is a known matrix and $\boldsymbol{m}_o$ is a known vector, and are used to describe the uncertainty regarding the initial state of the system. The matrix $\mathbf{A}(k)+\mathbf{A}_{\Delta}(k)$ models the uncertain dynamics of the system, at time k, with $\mathbf{A}(k)$ known and $\mathbf{A}_{\Delta}(k)$ uncertain, as further described in the sequel. A similar structure is assumed to the input and output matrices, i.e., $\mathbf{B}(k)+\mathbf{B}_{\Delta}(k)$ and $\mathbf{C}(k)+\mathbf{C}_{\Delta}(k)$, respectively. The matrix $\mathbf{L}_d(k)$ describes the direction upon which the disturbances can act, at time k, and is also assumed known. Moreover, it is assumed that the matrices of the dynamics of (1) are uniformly bounded, i.e., there exists $\alpha<\infty$ such that

$$\|\mathbf{A}(k)\|,\|\mathbf{A}_{\Delta}(k)\|,\|\mathbf{L}_d(k)\|,\|\mathbf{B}(k)\|,\|\mathbf{B}_{\Delta}(k)\|,\|\mathbf{C}(k)\|,\|\mathbf{C}_{\Delta}(k)\|\le\alpha,$$

for all k, where, for a matrix $\mathbf{M}$, $\|\mathbf{M}\|$ denotes the maximum singular value of $\mathbf{M}$. The elements of vector $\boldsymbol{v}(k)$ are represented as $v_i(k)$, so that $\boldsymbol{v}(k)=[v_1(k),\, v_2(k),\dots,v_m(k)]^{\mathrm{T}}$. The concatenation of a sequence of vectors $\boldsymbol{v}(k)$, $\boldsymbol{v}(k-1)$, $\cdots$, $\boldsymbol{v}(k-N+1)$ is denoted by

$$\boldsymbol{v}_N=\begin{bmatrix}\boldsymbol{v}(k)\\ \vdots\\ \boldsymbol{v}(k-N+1)\end{bmatrix}.$$

For the sake of simplicity, $\boldsymbol{v}$ is used instead of $\boldsymbol{v}_N$ whenever N can be inferred from the context. Also, for a matrix $\mathbf{M}\in\mathbb{R}^n$, let

$$\begin{bmatrix}\mathbf{M}\\ \star\end{bmatrix} := \begin{bmatrix}\mathbf{M}\\ -\mathbf{M}\end{bmatrix}.$$

Furthermore, it is assumed that

$$\mathbf{A}_{\Delta}(k)=\mathbf{A}_1(k)\Delta_1(k)+\mathbf{A}_2(k)\Delta_2(k)+\cdots+\mathbf{A}_{n_A}\Delta_{n_A}(k),$$

for $|\Delta_i(k)|\le 1, i=1,\dots,n_A$. The scalars $\Delta_i(k)$, $i=\{1,\dots,n_A\}$, represent parametric uncertainties, while the matrices $\mathbf{A}_i(k)$, $i=\{1,\dots,n_A\}$, are the directions which those uncertainties act upon. For the time being, it is assumed that

$$\mathbf{B}_{\Delta}(k)\equiv 0,\quad \mathbf{C}_{\Delta}(k)\equiv 0,\quad k\ge 0.$$

This assumption will be dropped in the sequel, and is considered here just for the sake of clarity.

The goal here is to find $\boldsymbol{x}(k+1)$, based upon (1) and with the additional knowledge that $\boldsymbol{x}(k)\in X(k)$,$\boldsymbol{x}(k-1)\in X(k-1)$, $\dots$,$\boldsymbol{x}(k-N)$ $\in$ $X(k-N)$ for some finite N. It is further required that for all $\boldsymbol{x}\in X(k+1)$, there exists $\boldsymbol{x}(k)\in X(k)$ such that the observations are compatible with (1). In other words, $X(k+1)$ should be the smallest set containing all the solutions to (1). A procedure for discrete time-varying linear systems was introduced in Shamma and Tu (1999), and preliminary results of the extension of this technique to uncertain plants were presented in Rosa et al. (2009, 2010).

However, for plants with uncertainties, the set $X(k+1)$ is, in general, non-convex, even if $X(k)$ is convex. Thus, it cannot be represented by a linear inequality as in (2). The approach suggested in Rosa et al. (2009), which assumes that $\mathrm{rank}(\mathbf{A}_i(k))=1$ for all $i\in\{1,\dots,n_A\}$, is to overbound this set by a convex polytope, therefore, adding some conservatism to the solution.

It is presented, in the sequel, a different approach, which does not require the rank assumption on the $\mathbf{A}_i(k)$ matrices, and that reduces to the procedure in Rosa et al. (2009) whenever such a rank condition is verified. This approach amounts to solving (1) for the vertices of the hyper-cube defined by $|\Delta_i(j)|\le 1, i=1,\dots,$ n_A, and $j=k-N+1,\dots,k$, as explained next.

Indeed, consider a realization of (1) where $\Delta_i(j)=\Delta_i^*(j)$, $i=1,\dots,n_A$, and $j=k-N+1,\dots,k$, and denote by $\mathbf{A}_{\Delta^*}(j)$ the corresponding uncertainty maps, i.e.,

$$\mathbf{A}_{\Delta^*}(j)=\mathbf{A}_1(j)\Delta_1^*(j)+\mathbf{A}_2(j)\Delta_2^*(j)+\cdots+\mathbf{A}_{n_A}(j)\Delta_{n_A}^*(j).$$

Then, the technique in Shamma and Tu (1999) can be used to design an SVO which computes a set-valued estimate of the state of the plant, by noting that (1) with $\Delta_i(j)=\Delta_i^*(j), i=1,\dots,n_A$, and $j=k-N+1,\dots,k$, is equivalent to stating that there exist $\boldsymbol{x}(k+1),\dots,\boldsymbol{x}(k-N+1)$, $\boldsymbol{y}(k)$, and $\boldsymbol{d}(k),\dots,\boldsymbol{d}(k-N+1)$, such that,

$$\mathbf{P}(k)\begin{bmatrix}\boldsymbol{x}(k+1)\\ \boldsymbol{x}(k)\\ \boldsymbol{x}(k-1)\\ \vdots\\ \boldsymbol{x}(k-N+1)\\ \boldsymbol{d}(k)\\ \boldsymbol{d}(k-1)\\ \vdots\\ \boldsymbol{d}(k-N+1)\end{bmatrix}\le\begin{bmatrix}\mathbf{B}(k)\boldsymbol{u}(k)\\ \star\\ \mathbf{F}_k^1\mathbf{B}(k-1)\boldsymbol{u}(k-1)+\mathbf{B}(k)\boldsymbol{u}(k)\\ \star\\ \vdots\\ \mathbf{F}_k^{N-2}\mathbf{B}(k-N+1)\boldsymbol{u}(k-N+1)+\cdots+\mathbf{B}(k)\boldsymbol{u}(k)\\ \star\\ \mathbf{1}\\ \vdots\\ \mathbf{1}\\ \tilde{\boldsymbol{m}}(k)\\ \vdots\\ \boldsymbol{m}(k-N)\end{bmatrix}$$

$$=: \boldsymbol{p}(k), \tag{3}$$

where

$$\mathbf{P}(k) = \begin{bmatrix}
\mathbf{I} & -\mathbf{F}_k^0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{L}_k & 0 & 0 & \cdots & 0 \\
\star & \star & \star & \star & \cdots & \star & \star & \star & \star & \cdots & \star \\
\mathbf{I} & \mathbf{0} & -\mathbf{F}_k^1 & \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{L}_k & -\mathbf{F}_k^0\mathbf{L}_{k-1} & \mathbf{0} & \cdots & \mathbf{0} \\
\star & \star & \star & \star & \cdots & \star & \star & \star & \star & \cdots & \star \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{F}_k^{N-1} & -\mathbf{L}_k & -\mathbf{F}_k^0\mathbf{L}_{k-1} & -\mathbf{F}_k^1\mathbf{L}_{k-2} & \cdots & -\mathbf{F}_k^{N-2}\mathbf{L}_{k-N+1} \\
\star & \star & \star & \star & \cdots & \star & \star & \star & \star & \cdots & \star \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\star & \star & \star & \star & \cdots & \star & \star & \star & \star & \cdots & \star \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \\
\star & \star & \star & \star & \cdots & \star & \star & \star & \star & \cdots & \star \\
\tilde{\mathbf{M}}_{k+1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \tilde{\mathbf{M}}_k & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \tilde{\mathbf{M}}_{k-N+1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0}
\end{bmatrix},$$

$\tilde{\mathbf{M}}_k = \begin{bmatrix} \mathbf{C}(k) \\ -\mathbf{C}(k) \end{bmatrix}$, $\tilde{\mathbf{m}}_k = \begin{bmatrix} \mathbf{1}\cdot\overline{n}+\boldsymbol{y}(k) \\ \mathbf{1}\cdot\overline{n}-\boldsymbol{y}(k) \end{bmatrix}$, $\mathbf{M}_k$ and $\boldsymbol{m}_k$ are defined such that $X(k) = \text{Set}(\mathbf{M}_k,\boldsymbol{m}_k)$, $\mathbf{L}_k = \mathbf{L}_d(k)$ and

$$\mathbf{F}_k^m := \mathbf{F}_k^m(\boldsymbol{\Delta}^*) := (\mathbf{A}(k)+\mathbf{A}_{\Delta^*}(k))\cdots(\mathbf{A}(k-m)+\mathbf{A}_{\Delta^*}(k-m)).$$

Notice that (3) can be obtained from the constraints on the disturbances and on the initial state, and from the relations

$$\begin{cases}
\boldsymbol{y}(k+1) = \mathbf{C}(k)\boldsymbol{x}(k+1)+\boldsymbol{n}(k+1), \\
\boldsymbol{x}(k+1) = \mathbf{F}_k^0(\boldsymbol{\Delta})\boldsymbol{x}(k)+\mathbf{L}_d(k)\boldsymbol{d}(k), \\
\boldsymbol{x}(k+1) = \mathbf{F}_k^1(\boldsymbol{\Delta})\boldsymbol{x}(k-1)+\mathbf{L}_d(k-1)\boldsymbol{d}(k-1)+\mathbf{L}_d(k)\boldsymbol{d}(k), \\
\vdots \\
\boldsymbol{x}(k+1) = \mathbf{F}_k^{N-1}\boldsymbol{x}(k-N+1)+\cdots+\mathbf{L}_d(k)\boldsymbol{d}(k).
\end{cases}$$

For given $\Delta_i^*(j)$, $i = 1,\ldots,n_A$, and $j = k-N+1,\ldots,k$, the *Fourier–Motzkin elimination method* (see Keerthi & Gilbert, 1987; Shamma & Tu, 1999) can be used to reduce (3) to an inequality of the form $\boldsymbol{M}_{\Delta^*}\boldsymbol{x}(k+1) \le \boldsymbol{m}_{\Delta^*}$. To see this, note that

$$[\boldsymbol{x}^{\mathrm{T}}(k+1),\ldots,\boldsymbol{x}^{\mathrm{T}}(k-N+1),\ \boldsymbol{d}^{\mathrm{T}}(k),\ldots,\boldsymbol{d}^{\mathrm{T}}(k-N+1)]^{\mathrm{T}} \in \mathbb{R}^{n+(n+n_d)N},$$

which means that (3) is a polytope defined in $\mathbb{R}^{n+(n+n_d)N}$. Thus, by projecting this polytope onto the subspace of the first n coordinates, one obtains a description of all the admissible $\boldsymbol{x}(k+1)$, which does not depend upon $\boldsymbol{x}(k),\ldots,\boldsymbol{x}(k-N+1)$, nor $\boldsymbol{d}(k),\ldots,\boldsymbol{d}(k-N+1)$, as depicted in Fig. 1.

Let $\boldsymbol{v}_i$, for $i = 1,\ldots,2^{Nn_A}$, denote a vertex of the hyper-cube

$$\mathcal{C} := \{\boldsymbol{\delta} \in \mathbb{R}^{Nn_A} : \|\boldsymbol{\delta}\| \le 1\}, \tag{4}$$

where $\boldsymbol{v}_i = \boldsymbol{v}_j \Leftrightarrow i = j$. Then, denote by $\hat{X}_{\boldsymbol{v}_i}(k+1)$ the set of points $\boldsymbol{x}(k+1)$ that satisfy (3) with $\mathbf{A}_{\Delta^*} = \mathbf{A}_{\boldsymbol{v}_i}$ and with $\boldsymbol{x}(k) \in \hat{X}(k)$. The notation $\mathbf{A}_\Delta = \mathbf{A}_{\Delta^*}$ is used to indicate that $\mathbf{A}_\Delta(m) = \mathbf{A}_{\Delta^*}(m)$ for $m = k-N+1,\ldots,k$, and $\mathbf{A}_{\Delta^*} = \mathbf{A}_{\boldsymbol{v}_i}$ to indicate that

$$\Delta^* = \begin{bmatrix} \Delta^*(k) \\ \vdots \\ \Delta^*(k-N+1) \end{bmatrix} = v_i.$$

Further define

$$\hat{X}(k+1) : = \text{co}\{\hat{X}_{v_1}(k+1),\hat{X}_{v_2}(k+1),\ldots,\hat{X}_{v_{(2^{Nn_A})}}(k+1)\},$$

where $\text{co}\{\boldsymbol{p}_1,\ldots,\boldsymbol{p}_m\}$ is the smallest convex set containing the points $\boldsymbol{p}_1,\ldots,\boldsymbol{p}_m$, also known as the convex hull of $\boldsymbol{p}_1,\ldots,\boldsymbol{p}_m$.

Since $X(k+1)$ can be, in general, non-convex even if $X(k)$ is convex, $\hat{X}(k+1)$ is used to overbound the set $X(k+1)$. As seen in the sequel, $\hat{X}(k+1)$ is an estimate of $X(k+1)$ which contains the latter set. An illustration of $\hat{X}(k)$ for n_A=1, i.e., for a single uncertainty, and for N=1 is depicted in Fig. 2.

One advantage of using $\hat{X}(k+1)$ instead of another convex set is stated in the following proposition.

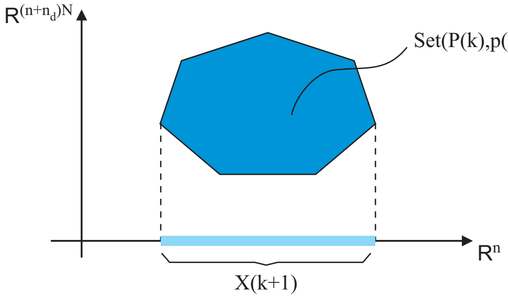

Fig. 1. Projection of Set($\mathbf{P}(k)$,$\boldsymbol{p}(k)$) onto $\mathbb{R}^n$.

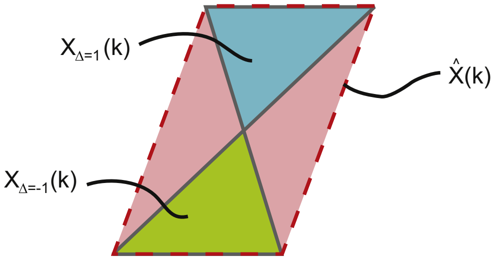

Fig. 2. Convex hull, $\hat{X}(k)$, of the sets generated by the solutions to (1) with n_A=1, N=1, and for $\Delta = 1$ and $\Delta = -1$.

Proposition 1. *Suppose that a system described by (1) with $\boldsymbol{x}(0) = \boldsymbol{x}_o$ and $\boldsymbol{u}(k) = \boldsymbol{0},\forall k$, and where the matrices of the dynamics are uniformly bounded, satisfies, for sufficiently large N^*,*

$$\gamma_N := \max_{\substack{\boldsymbol{\Delta}(k),\ldots,\boldsymbol{\Delta}(k+N) \\ \|\boldsymbol{\Delta}(m)\| \le 1,\forall m \\ k \ge 0}} \|\mathbf{F}_k^{N-1}(\Delta)\| < 1,$$

for all $N \ge N^*$. Then, $\hat{X}(k)$ cannot grow without bound.

Proof. Consider a sequence of hyper-cubes, denoted by $\Psi(1)$, $\Psi(2),\ldots,\Psi(m)$, that contain the sets $\hat{X}(1)$,$\hat{X}(2),\ldots,\hat{X}(m)$, respectively. Let $N \ge N^*$. Then, an SVO can be synthesized to generate such sets $\Psi(1),\Psi(2),\ldots,\Psi(m)$, using the following inequality:

$$\|\boldsymbol{x}(k+N)\| \le \gamma_N\|\boldsymbol{x}(k)\|+\delta_N, \tag{5}$$

where

$$\delta_N = \max_{\substack{\boldsymbol{d}(k),\ldots,\boldsymbol{d}(k+N-1) \\ \boldsymbol{\Delta}(k),\ldots,\boldsymbol{\Delta}(k+N)}} \|\mathbf{F}_k^{N-2}\mathbf{L}_d(k)\boldsymbol{d}(k)+\cdots+\mathbf{L}_d(k+N-1)\boldsymbol{d}(k+N-1)\|.$$

Notice that it suffices to show that the sequence $\Psi(1),\Psi(2),\ldots$, $\Psi(m)$ does not grow without bound, since these terms are the smallest hypercubes that contain $\hat{X}(1)$,$\hat{X}(2),\ldots$,$\hat{X}(m)$. Given that $\gamma_N < 1$ by assumption and that $\|\delta_N\| < \infty$ since $\|\boldsymbol{d}\| < \infty$ and given that the matrices of the dynamics are uniformly bounded, one concludes that the sets defined by (5) cannot grow without bound. □

Remark 1. Notice that, in order to guarantee that $\hat{X}$ does not grow without bound, an SVO should use the N most recent estimates. In other words, the estimation of $\hat{X}(k+1)$ should take into account the estimates $\hat{X}(k)$,$\hat{X}(k-1),\ldots,\hat{X}(k-N+1)$.

Another advantage of the use of set $\hat{X}$ stems from the fact that $X(k) \subseteq \hat{X}(k)$. This is stated in the following proposition:

Proposition 2. *Consider a system described by (1) with the aforementioned constraints, and assume that $X(0) \subseteq \hat{X}(0)$. Then $X(k) \subseteq \hat{X}(k)$ for all $k \in \{0,1,2,\ldots\}$.*

Proof. Denote by $X_{\boldsymbol{v}_i}(k+1)$ the set of points $\boldsymbol{x}(k+1)$ that satisfy (1) with the aforementioned constraints, and with $[\boldsymbol{\Delta}^{\mathrm{T}}(k),\ldots,\boldsymbol{\Delta}^{\mathrm{T}}(k-N+1)]^{\mathrm{T}} = \boldsymbol{v}_i$ and $\boldsymbol{x}(k) \in X(k),\ldots,\boldsymbol{x}(k-N+1) \in X(k-N+1)$.

Further define

$$X^{*}(k+1) := \mathrm{co}\{X_{\mathbf{v}_1}(k+1), X_{\mathbf{v}_2}(k+1), \ldots, X_{\mathbf{v}_{2^{(Nn_\Delta)}}}(k+1)\}.$$

Then, it is clear that $X^{*}(k+1) \subseteq \hat{X}(k+1)$ if $X(j) \subseteq \hat{X}(j)$, for $0 \le j \le k$. By induction, it is straightforward to conclude that $X^{*}(i) \subseteq \hat{X}(i)$ for all $i \in \{0,1,2,\ldots\}$, since it was assumed that $X(0) \subseteq \hat{X}(0)$.

Hence, the only thing left to prove is that $X(k) \subseteq X^{*}(k)$. To see this, note that the disturbances and control input in (1) do not add any non-convexities to the set-valued estimate of the state at each sampling time. Therefore, without loss of generality, assume that $\boldsymbol{d}(\cdot) \equiv 0$ and $\boldsymbol{u}(\cdot) \equiv 0$ in (1). Let $\boldsymbol{x}^{\star} \in X(k)$ be obtained by using a vector $\boldsymbol{\Delta} \in \mathcal{C}$ in (4). Thus, it can be defined as

$$\boldsymbol{x}^{\star} := \left[\prod_{i=1}^{N} (\mathbf{A}_o(k-i) + \alpha_i \mathbf{A}(k-i) - (1-\alpha_i)\mathbf{A}_1(k-i))\right] \boldsymbol{x}, \tag{6}$$

where $\boldsymbol{x} \in X(k-N)$, $0 \le \alpha_i \le 1$, for $i \in \{1,\ldots,N\}$, and assuming a single uncertainty, $|\Delta(i)| \le 1$, $i \ge 0$, for the sake of simplicity. (Recall that, in this scenario, $\boldsymbol{\Delta} = [\Delta^{\mathrm{T}}(k-1) \cdots \Delta^{\mathrm{T}}(k-N)]^{\mathrm{T}}$). It will now be shown that $\boldsymbol{x}^{\star}$ can be obtained as a convex combination of points satisfying (1), with $\boldsymbol{d}(\cdot) \equiv 0$, $\boldsymbol{u}(\cdot) \equiv 0$, $\boldsymbol{x}(k-N) \in X(k-N)$, and for $\boldsymbol{\Delta}$ in the vertices of $\mathcal{C}$. Define

$$\begin{cases} \mathbf{A}^{+}(k) = \mathbf{A}(k) + \mathbf{A}_1(k), \\ \mathbf{A}^{-}(k) = \mathbf{A}(k) - \mathbf{A}_1(k), \end{cases}$$

and let

$$\begin{cases} \boldsymbol{x}_1^1 = \mathbf{A}^{+}(k-1)\mathbf{A}^{+}(k-2)\cdots\mathbf{A}^{+}(k-N)\boldsymbol{x}, \\ \boldsymbol{x}_2^1 = \mathbf{A}^{-}(k-1)\mathbf{A}^{+}(k-2)\cdots\mathbf{A}^{+}(k-N)\boldsymbol{x}, \\ \boldsymbol{x}_3^1 = \mathbf{A}^{+}(k-1)\mathbf{A}^{-}(k-2)\cdots\mathbf{A}^{+}(k-N)\boldsymbol{x}, \\ \boldsymbol{x}_4^1 = \mathbf{A}^{-}(k-1)\mathbf{A}^{-}(k-2)\cdots\mathbf{A}^{+}(k-N)\boldsymbol{x}, \\ \vdots \\ \boldsymbol{x}_{2^N-1}^1 = \mathbf{A}^{+}(k-1)\mathbf{A}^{-}(k-2)\cdots\mathbf{A}^{-}(k-N)\boldsymbol{x}, \\ \boldsymbol{x}_{2^N}^1 = \mathbf{A}^{-}(k-1)\mathbf{A}^{-}(k-2)\cdots\mathbf{A}^{-}(k-N)\boldsymbol{x}, \end{cases}$$

where $\boldsymbol{x} \in X(k-N)$. Now let

$$\begin{cases} \boldsymbol{x}_1^2(\gamma^1) = [(\mathbf{A}(k-1) + \gamma^1\mathbf{A}_1(k-1) - (1-\gamma^1)\mathbf{A}_1(k-1))\mathbf{A}^{+}(k-2)\cdots\mathbf{A}^{+}(k-N)]x, \\ \boldsymbol{x}_2^2(\gamma^1) = [(\mathbf{A}(k-1) + \gamma^1\mathbf{A}_1(k-1) - (1-\gamma^1)\mathbf{A}_1(k-1))\mathbf{A}^{-}(k-2)\cdots\mathbf{A}^{+}(k-N)]x, \\ \vdots \\ \boldsymbol{x}_{2^{N-1}}^2(\gamma^1) = [(\mathbf{A}(k-1) + \gamma^1\mathbf{A}_1(k-1) - (1-\gamma^1)\mathbf{A}_1(k-1))\mathbf{A}^{-}(k-2)\cdots\mathbf{A}^{-}(k-N)]x. \end{cases}$$

Notice that $\boldsymbol{x}_1^2$ is a convex combination of $\boldsymbol{x}_1^1$ with $\boldsymbol{x}_2^1$, while $\boldsymbol{x}_2^2$ is a convex combination of $\boldsymbol{x}_3^1$ with $\boldsymbol{x}_4^1$, and so on. Straightforward but tedious calculations show that

$$\boldsymbol{x}_1^{N+1}(\gamma^1,\ldots,\gamma^N) = \left[\prod_{i=1}^{N} (\mathbf{A}(k-i) + \gamma^i \mathbf{A}_1(k-i) - (1-\gamma^i)\mathbf{A}_1(k-i))\right] \boldsymbol{x}. \tag{7}$$

Hence, by comparing (6) with (7), one gets

$$\boldsymbol{x}_1^{N+1}(\gamma^1,\ldots,\gamma^N) = \boldsymbol{x}^{\star},$$

for $\gamma^i = \alpha_i$, for all $i \in \{1,\ldots,N\}$. A similar procedure can be used for the case where more than one uncertainty is considered, thus concluding the proof. □

The uncertainty in $\mathbf{B}_\Delta(k)$ may also be modeled as an exogenous disturbance, since $\mathbf{B}(k)$ and $\boldsymbol{u}(k)$ in the model are assumed known at each sampling time. In particular, if the model is described by

$$\boldsymbol{x}(k+1) = \mathbf{A}(k)\boldsymbol{x}(k) + (\mathbf{B}(k) + \mathbf{B}_1(k)\Delta(k))\boldsymbol{u}(k),$$

where $\mathbf{B}_1(k)$ is a known time-varying matrix and where $\Delta(k) \in \mathbb{R}$ with $|\Delta(k)| \le 1$, then it can be rewritten as

$$\boldsymbol{x}(k+1) = \mathbf{A}(k)\boldsymbol{x}(k) + \mathbf{B}(k)\boldsymbol{u}(k) + \mathbf{L}_d(k)\boldsymbol{d}(k),$$

where $\mathbf{L}_d(k) = \mathbf{B}_1(k)\boldsymbol{u}(k)$ and $\boldsymbol{d}(k)$ is an exogenous disturbance with $\|\boldsymbol{d}\| \le 1$.

The uncertainty in $\mathbf{C}_\Delta(k)$ requires a more detailed analysis. Indeed, consider a dynamic system, S, described by

$$S : \begin{cases} \boldsymbol{x}(k+1) = \mathbf{A}(k)\boldsymbol{x}(k) + \mathbf{B}(k)\boldsymbol{u}(k) + \mathbf{L}(k)\boldsymbol{d}(k), \\ \boldsymbol{y}(k) = \left(\mathbf{C}_o(k) + \sum_{j=1}^{n_\Delta} \Delta_j(k)\mathbf{C}_j(k)\right)\boldsymbol{x}(k) + \boldsymbol{n}(k), \end{cases} \tag{8}$$

with the aforementioned constraints, and $\boldsymbol{\Delta}(k) \in \mathbb{R}^{n_\Delta}$. It is also assumed that

$$|\Delta_j(k)| \le 1.$$

Notice that S is equivalent to

$$S \equiv (\bar{S}_o + \bar{\boldsymbol{n}}) + \sum_{j=1}^{n_\Delta} (\Delta_j \bar{S}_j + \bar{\boldsymbol{n}}), \tag{9}$$

where, for $j \in \{1,\ldots,n_\Delta\}$,

$$\bar{S}_j : \begin{cases} \boldsymbol{x}_j(k+1) = \mathbf{A}(k)\boldsymbol{x}_j(k) + \mathbf{B}(k)\boldsymbol{u}(k) + \mathbf{L}(k)\boldsymbol{d}(k), \\ \boldsymbol{y}_j(k) = \mathbf{C}_j(k)\boldsymbol{x}_j(k), \end{cases}$$

with $\boldsymbol{x}_j(0) = \boldsymbol{x}(0)$ for all $j \in \{0,\ldots,n_\Delta\}$, and $\bar{\boldsymbol{n}} = \boldsymbol{n}/(n_\Delta + 1)$. The block diagram of (9) is depicted in Fig. 3.

Since each $\bar{S}_j$, for $j \in \{0,\ldots,n_\Delta\}$, is a linear system, and each $\Delta_j(k)$, for $j \in \{1,\ldots,n_\Delta\}$ and $k \ge 0$, is an uncertain scalar, one obtains

$$S \equiv (\bar{S}_o + \bar{\boldsymbol{n}}) + \sum_{j=1}^{n_\Delta} (\tilde{S}_j + \bar{\boldsymbol{n}}), \tag{10}$$

where

$$\tilde{S}_j : \begin{cases} \boldsymbol{x}_j(k+1) = \mathbf{A}(k)\boldsymbol{x}_j(k) + \mathbf{B}(k)\Delta_j(k)\boldsymbol{u}(k) + \mathbf{L}(k)\Delta_j(k)\boldsymbol{d}(k), \\ \boldsymbol{y}_j(k) = \mathbf{C}_j(k)\boldsymbol{x}_j(k). \end{cases}$$

Notice that (10) describes an LPV system with uncertain input. Nevertheless, the exogenous disturbances are now multiplied by the uncertainties $\Delta_j(k)$, and hence $\tilde{S}_j$ depends upon $\Delta_j(k)$ and $\boldsymbol{d}(k)$ in a bilinear fashion. However, this can be avoided by introducing the following relaxation. Since $|\Delta_j(k)| \le 1$,

$$\tilde{\boldsymbol{d}}_j(k) := \Delta_j(k)\boldsymbol{d}(k) \Rightarrow \|\tilde{\boldsymbol{d}}_j(k)\| \le \|\boldsymbol{d}(k)\|. \tag{11}$$

Thus, by substituting $\Delta_j(k)\boldsymbol{d}(k)$ in (10) by $\tilde{\boldsymbol{d}}_j(k)$ as in (11), one obtains a description of the system which is linear in the unknown variables, at the cost of some conservatism due to the implication in (11).

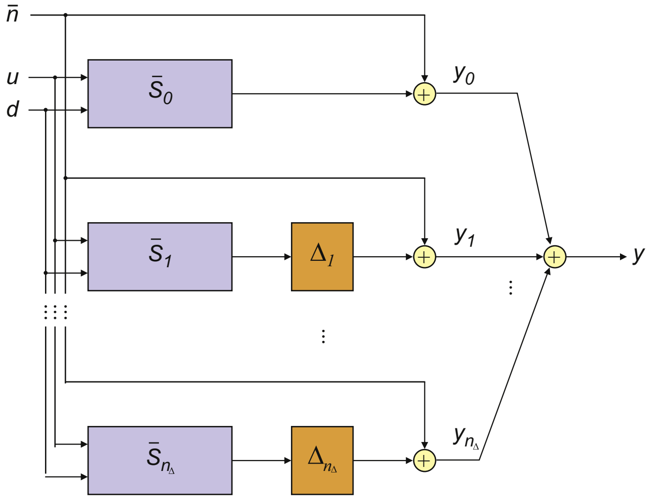

Fig. 3. Block diagram of an LPV system with uncertainties in the output.

3. Fault detection using SVOs

In this section, an SVO-based method to detect faults is introduced. Since a fault can be defined as a deviation of the plant dynamics from the nominal model, an SVO can be used to detect such a mismatch between the predicted and the actual output of the system. As further stressed in the sequel, this nominal model of the plant must account for unmodeled dynamics, exogenous disturbances and measurement noise.

The following proposition is used to detect faulty behaviors of plants that are modeled by systems represented by (1).

Proposition 3. *Consider a non-faulty plant described by* (1) *and a corresponding SVO, as introduced in Section* 2*. Then, if* $\hat{X}(k)=\emptyset$ *for some* $k\geq 0$*, a fault has occurred at some time* k_o*, where* $k_o\leq k$.

Proof. If $\hat{X}(k)$ is empty for some k, then the observations are not compatible with the model of the plant. Since it was assumed that the non-faulty plant can be described by (1), one concludes that a fault has occurred. □

Using Proposition 3, the architecture depicted in Fig. 4 can be used to tackle the problem of fault detection for discrete-time LPV systems. Notice that the FD filter is composed of an SVO and a logic block, which decides whether a fault is diagnosed or not, according to the emptiness or not of the set-valued estimate of the state at each sampling time.

3.1. The (in)distinguishability problem

The sufficient condition in Proposition 3 guarantees that there will be no false alarms, unless model (1) does not properly describe the nominal plant. It does not, however, provide any guarantees in terms of fault detection, after a certain fault has occurred in the actual system. Indeed, it may happen that a fault is not detected using the approach presented herein. Assuming that (1) adequately describes the plant model, there are two reasons that can justify a missed detection of a fault using the proposed technique:

1. The conservatism added due to the convexification of the set of admissible state values, can lead to the validation of certain observations, that are in fact not compatible with (1).
2. The input/output data sequence generated by the faulty plant can, under certain circumstances, be generated also by the nominal plant, for the prespecified level of state disturbances, model uncertainty, and measurement noise. If such an event can never occur, it is said that the faulty and the nominal plant are *distinguishable*.

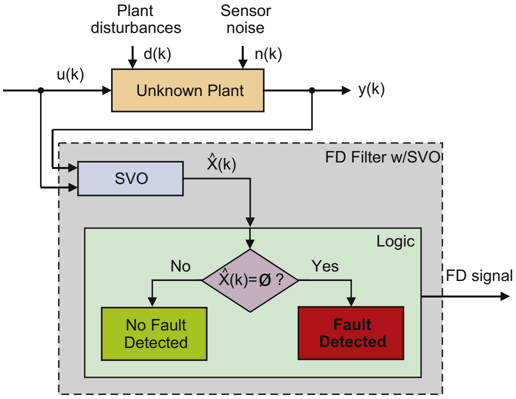

Fig. 4. Fault detection (FD) architecture for uncertain plants using a set-valued observer (SVO).

These topics are still under research and some preliminary results are presented in Rosa and Silvestre (2011). The interested reader is further referred to Grewal and Glover (1976), DiStefano (1977), DiStefano and Cobelli (1980), Walter, Lecourtier, and Happel (1984), Lou and Si (2009) and Johansson (2009).

Remark 2. Proposition 3 can only guarantee that there will be no false alarms, as long as the bounds on the exogenous disturbances are not violated. However, in several applications, such an assumption may not be plausible from a practical point of view, and thus the probability of violating such bounds can be used as an upper bound on the likelihood of having false alarms.

4. Fault isolation using SVOs

The fault isolation techniques available in the literature aim at identifying a very precise faulty behavior, after a general fault is detected. This means that, unlike an FD filter, an FDI filter should not only be able to detect a faulty behavior of a plant, but also to provide information regarding its location. In particular, FDI filters should be able to separate between the three broad types of failures enumerated in the sequel.

The proposed technique is also suitable for fault isolation, as long as the corresponding model of the fault is considered during the design of the SVOs. The main idea in this case is to resort to model invalidation, as follows (see Rosa, Silvestre, & Athans, 2011, for further details on *model falsification* using SVOs). As an example, suppose that there are three possible faulty models, $M\#1$, $M\#2$ and $M\#3$, for a given plant, plus a nominal model, $M\#4$. The goal is to decide which model (if any) is able to justify the input/output data that one is obtaining from the sensors and actuators' commands. Therefore, assume that, at a given initial time, t_o, all the four models are plausible, as depicted in Fig. 5. Further suppose that, at time t_1, model $M\#4$ is invalidated, i.e., the sensors readings cannot be explained by model $M\#4$. Hence, since this is the nominal model, one concludes that a fault has occurred.

Moreover, consider that, at time t_2, model $M\#2$ is invalidated and that, finally, model $M\#1$ is invalidated at time t_3. Then, one concludes that the only model capable of explaining the input/output time-series generated by the plant is model $M\#3$. Thus, a fault was properly detected at time t_1, and isolated at time t_3.

In this paper, an alternative to the model invalidation methods in Poolla, Khargonekar, Tikku, Krause, and Nagpal (1994) and Bianchi and Sánchez-Peña (2010) is proposed, that compares the whole input/output sequence generated by the plant up to a given time instant, with the dynamic model of the plant. Contrary to the aforementioned solutions, the technique presented herein relies on a recursive algorithm, which is, in general, more suitable for real-time implementation.

4.1. Architecture I

Consider, for instance, a loss-of-effectiveness type of fault in one actuator. As described in the sequel, this fault can be modeled by multiplying the actuator input signal by a constant $\lambda\in[0,1]$. Therefore, consider an SVO, as described in Section 2, designed for

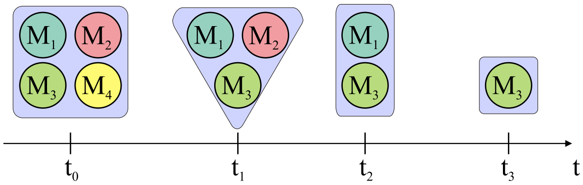

Fig. 5. Example of the time-evolution of a set of models that are able to describe the input/output behavior of a given plant.

a plant with this type of uncertainty. Then, such an SVO would validate observations from a model with any value of $\lambda \in [0,1]$ and, in particular, for $\lambda = 1$, which corresponds to the nominal plant.

Indeed, the architecture described in the sequel, referred to as Architecture I and depicted in Fig. 6(a), assumes that the fault isolation filters provide valid set-valued estimates for the state of plant, not only for the faulty plant, with a specific fault, but also for the non-faulty plant. In addition, one SVO for the non-faulty (probably uncertain and time-varying) plant – referred to as *nominal SVO* – can be used to detect the fault. As described in Section 3, the set-valued estimate for the state of the plant, obtained from this observer, is non-empty, if the plant does not present a faulty behavior. If the state estimate of the nominal SVO is the empty set, a fault has occurred. A fault is completely isolated whenever a single fault isolation filter has a non-empty set-valued state estimation.

However, under certain circumstances, it may not be convenient to assume that the dynamic systems modeled by the fault isolation filters include the nominal model of the plant. As an example, suppose that one is interested in detecting faults $F_1, F_2, \ldots, F_N$. If Architecture I is used, one FD filter and N fault isolation filters are required. Each of these fault isolation filters computes two set-valued estimates for the state of the plant: one for the non-faulty model, $\hat{X}_{\text{nom}}$; and one for the faulty plant, with the corresponding failure, $\hat{X}_{\text{failure\#i}}$. Moreover, the SVOs require these set-valued estimates to be convex regions. Hence, the set-valued estimate of each fault isolation filter can be written as $\hat{X}_i = \text{co}\{\hat{X}_{\text{nom}}, \hat{X}_{\text{failure\#i}}\}$. This can add conservatism to the solution, and thus the isolation of the faults may become more difficult. A solution to overcome this problem is presented in the sequel.

4.2. Architecture II

The architecture described in what follows, referred to as Architecture II, requires two additional SVOs, besides the faults isolation SVOs, namely:

1. one SVO for the non-faulty (probably uncertain and time-varying) plant—referred to as *nominal SVO*;
2. another SVO – referred to as *Global SVO* – providing set-valued estimates of the state, which are valid not only for the non-faulty plant, but also for the faulty plant.

The *nominal SVO* is used for fault detection only. Notice that this SVO produces state estimates valid not only for the nominal model of the plant, but also for any plant belonging to the family of admissible plants. If the state estimate of this SVO is the empty set, a fault has occurred. Hence, the fault isolation SVOs are initialized with the state estimate of the *Global SVO*. A fault is completely isolated whenever a single fault isolation SVO has a non-empty set-valued state estimation.

The proposed FDI architecture is depicted in Fig. 6(b). It should be stressed that the FD filters that are designed for specific faults are only initialized with the set-valued state estimate of the Global SVO when they are signaled by the nominal FD filter that a fault has occurred.

Architecture II is a general approach which, however, may lead to some practical problems, since the set-valued state estimate of the aforementioned *Global SVO* can be very *large*, which, in turn, can increase the fault isolation period.

4.3. Types of faults

The focus of this paper is on three broad classes of failures that can be found in the literature, namely system dynamics failures, actuator failures, and sensor failures. It is presented, in the sequel, an outline of the description of each of the aforementioned classes of failures.

4.3.1. System dynamics failures

System dynamics failures can be used to describe significant changes in the model of the plant that can be in general interpreted as changes in one or more elements of matrix $\mathbf{A}$ of the dynamic system in (1). These changes can be modeled by the parametric uncertainties in (1) and, thus, the SVOs can be used to detect and isolate such a class of faults.

Typical examples of such failures include icing and broken surfaces in aircrafts (Esteban, 2004).

4.3.2. Actuators failures

Another important type of failures are those in the actuators. Such faults can, in general, be described by changes in matrix $\mathbf{B}$ in (1), and/or changes in the input signal, $\boldsymbol{u}(\cdot)$. Mathematically, this can

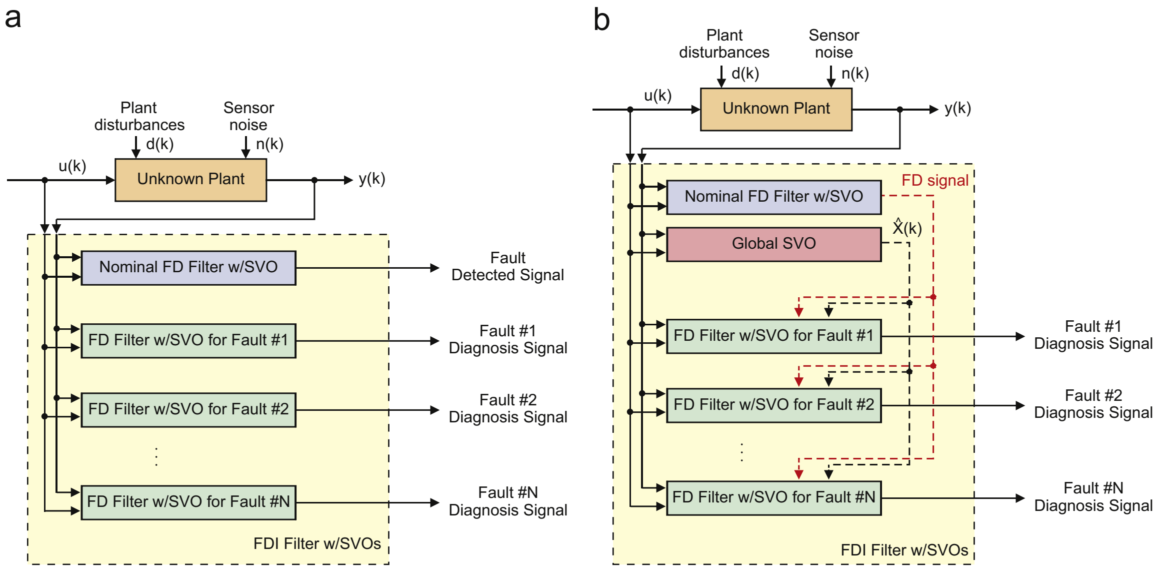
a
Plant disturbances
Sensor noise
d(k)
n(k)
u(k)
Unknown Plant
y(k)
Nominal FD Filter w/SVO
Fault Detected Signal
FD Filter w/SVO for Fault #1
Fault #1 Diagnosis Signal
FD Filter w/SVO for Fault #2
Fault #2 Diagnosis Signal
FD Filter w/SVO for Fault #N
Fault #N Diagnosis Signal
FDI Filter w/SVOs
b
Plant disturbances
Sensor noise
d(k)
n(k)
u(k)
Unknown Plant
y(k)
FD signal
Nominal FD Filter w/SVO
Global SVO
X̂(k)
FD Filter w/SVO for Fault #1
Fault #1 Diagnosis Signal
FD Filter w/SVO for Fault #2
Fault #2 Diagnosis Signal
FD Filter w/SVO for Fault #N
Fault #N Diagnosis Signal
FDI Filter w/SVOs

Fig. 6. Fault detection and isolation (FDI) architectures for uncertain LPV systems using set-valued observers (SVOs). (a) Architecture I. (b) Architecture II.

Table 1
Modeling of common actuator faults.

Type of fault	Modeling
Hard-over (saturation)	$\mathbf{M}(k)=\mathbf{B}(k)$ and $\boldsymbol{m}(k)=-\boldsymbol{u}(k)+\boldsymbol{u}_{\text{sat}}$
Loss-of-effectiveness	$\mathbf{M}(k)=\mathbf{B}(k)$ and $\boldsymbol{m}(k)=-\boldsymbol{u}(k)+\alpha\boldsymbol{u}(k), \alpha\in\mathbb{R}$
Floating	$\mathbf{M}(k)=\mathbf{B}(k)$ and $\boldsymbol{m}(k)=-\boldsymbol{u}(k)$
Bias (stuck)	$\mathbf{M}(k)=\mathbf{B}(k)$ and $\boldsymbol{m}(k)=-\boldsymbol{u}(k)+\beta, \beta\in\mathbb{R}$
Noise	$\mathbf{M}(k)=\mathbf{B}(k)$ and $\boldsymbol{m}(k)=$ noise

Table 2
Modeling of common sensor faults.

Type of fault	Modeling
Dead sensor	$\mathbf{Q}(k)=\mathbf{C}(k)$ and $\boldsymbol{q}(k)=-\boldsymbol{x}(k)$
Scale factor	$\mathbf{Q}(k)=\mathbf{C}(k)$ and $\boldsymbol{q}(k)=-\alpha\boldsymbol{x}(k),\ \alpha>0$
Biased sensor	$\mathbf{Q}(k)=\mathbf{C}(k)$ and $q(k)=\beta,\ \beta\in\mathbb{R}$
Random drift	$\mathbf{Q}(k)=\mathbf{C}(k)$ and $\boldsymbol{q}(k)=$ random variable

Table 3
Parameters of the UAV.

Parameter	Value
S	2.365 m^2
c	1.34 m
Aspect Ratio	2.047
m	39.7 kg
x_{cg}	1.217 m
I_y	10.43 kg m^2

be described by

$$\begin{cases}\boldsymbol{x}(k+1)=\mathbf{A}(k)\boldsymbol{x}(k)+\mathbf{A}_{\Delta}(k)\boldsymbol{x}(k)+\mathbf{L}_d(k)\boldsymbol{d}(k)+\mathbf{B}(k)\boldsymbol{u}(k)+\mathbf{M}(k)\boldsymbol{m}(k),\\ \boldsymbol{y}(k)=\mathbf{C}(k)\boldsymbol{x}(k)+\boldsymbol{n}(k),\end{cases} \tag{12}$$

where $\mathbf{M}(k)\boldsymbol{m}(k)$ accounts for the fault. It should be noticed that this model can describe a broad range of actuator faults, as summarized in Table 1 for a single control input plant—see Esteban (2004) and references therein for further details.

The model described by (12) is clearly compatible with (1) and, hence, the SVOs can also be used to detect and isolate actuator faults.

4.3.3. *Sensors failures*

Finally, sensors failures can also be treated in a similar manner. Consider the following description of the plant,

$$\begin{cases}\boldsymbol{x}(k+1)=\mathbf{A}(k)\boldsymbol{x}(k)+\mathbf{A}_{\Delta}(k)\boldsymbol{x}(k)+\mathbf{L}_d(k)\boldsymbol{d}(k)+\mathbf{B}(k)\boldsymbol{u}(k),\\ \boldsymbol{y}(k)=\mathbf{C}(k)\boldsymbol{x}(k)+\boldsymbol{n}(k)+\mathbf{Q}(k)\boldsymbol{q}(k),\end{cases} \tag{13}$$

where $\mathbf{Q}(k)\boldsymbol{q}(k)$ accounts for the sensor fault. Table 2 summarizes the most common sensor faults and corresponding models.

Similarly to what happened in the previous cases, the model described by (13) is also compatible with (1) and, therefore, the SVOs can be used for fault detection and isolation of sensor failures.

5. Aircraft longitudinal LPV model

One of the main objectives of this paper is to illustrate the applicability and performance of the FDI strategy introduced in the previous sections, to the model of the longitudinal dynamics of a fixed-wing aircraft. In particular, the Eclipse Unmanned Air Vehicle (UAV) described in Samy, Postlethwaite, and Gu (2011) and Chen (2010) is considered. Some of the properties of this type of UAV are defined in Table 3. It should be noticed that, for the implementation of the proposed FDI strategy, the dynamics of the plant must be described by an LPV model. Thus, this section will also address the problem of describing the longitudinal dynamics of the aircraft through an LPV model. Nevertheless, this LPV model is only used to design the SVOs. For the simulation results presented in Section 6, the full nonlinear model of the longitudinal dynamics of the aircraft is used instead.

The motion of the UAV can be described by a 6 degrees-of-freedom model, as follows:

$$\begin{cases}\dot{u}(t)=\dfrac{1}{m}X(t)-q(t)w(t),\\ \dot{w}(t)=\dfrac{1}{m}Z(t)-q(t)u(t),\\ \dot{q}(t)=\dfrac{M(t)}{I_y},\\ \dot{\theta}(t)=q(t),\\ \dot{P}_N=u(t)\cos(\theta(t))+w(t)\sin(\theta(t)),\\ \dot{h}=u(t)\sin(\theta(t))-w(t)\cos(\theta(t)),\end{cases} \tag{14}$$

where $X(\cdot)$, $Z(\cdot)$, and $M(\cdot)$ are the axial force, normal force, and pitching moment, respectively. The remaining notation has been widely used in the aerospace community (*cf.* Stevens & Lewis, 2003), although it should be noticed that $u(\cdot)$ refers to the forward airspeed, in the body frame, and should not be confused with the control input in (1). Furthermore, define the angle-of-attack (AOA) of the UAV as

$$\alpha(t)=\arctan\left(\frac{w(t)}{u(t)}\right),$$

and the airspeed as

$$V_{\text{AS}}(t)=\sqrt{u^2(t)+w^2(t)}.$$

The forces $X(\cdot)$ and $Z(\cdot)$, and moment $M(\cdot)$, are due to aerodynamic, engine thrust and gravitational effects. The aerodynamic effects are generated by the wings, fuselage, and surfaces of the UAV, and are functions of $\alpha(\cdot)$, $\dot{\alpha}(\cdot)$, $V_{\text{AS}}(\cdot)$, $q(\cdot)$, of the air density, and of the deflection of the control surfaces.

Thus, it is clear that the longitudinal dynamics of an aircraft are highly nonlinear, and depend on several (time-varying) parameters, such as the dynamic pressure and the aerodynamic coefficients. However, in wing-level flight, these dynamics are well-described by LPV models, which depend upon the airspeed, V_{AS}. In particular, using the derivations in Chen (2010), one obtains the following LPV description of the longitudinal dynamics of the Eclipse UAV:

$$\frac{d}{dt}\boldsymbol{x}(t)=\mathbf{A}(V_{\text{AS}})\boldsymbol{x}(t)+\mathbf{B}(V_{\text{AS}})\boldsymbol{\delta}(t), \tag{15}$$

where the state is defined as

$$\boldsymbol{x}(t)=\begin{bmatrix}u(t)\\ w(t)\\ \theta(t)\\ q(t)\end{bmatrix},$$

the control input is given by

$$\boldsymbol{\delta}(\boldsymbol{t})=\begin{bmatrix}\eta(t)\\ \tau(t)\end{bmatrix},$$

where $\eta(t)$ is the deflection of the elevator and $\tau(t)$ is proportional to the thrust force. Finally, the matrices of the dynamics are given by

$$\mathbf{A}(V_{\mathrm{AS}}) = \begin{bmatrix} X_u & X_w & X_\theta & X_q \\ Z_u & Z_w & Z_\theta & Z_q \\ 0 & 0 & 0 & 1 \\ M_u & M_w & M_\theta & M_q \end{bmatrix}, \quad \mathbf{B}(V_{\mathrm{AS}}) = \begin{bmatrix} X_\eta & X_\tau \\ Z_\eta & 0 \\ 0 & 0 \\ M_\eta & M_\tau \end{bmatrix}, \tag{16}$$

where

$X_u = -0.00102V_{\mathrm{AS}} - 0.009115$,
$X_w = 0.00018V_{\mathrm{AS}}^2 + 0.025V_{\mathrm{AS}} - 1.0499$,
$X_\theta = -9.67$,
$X_q = -0.00166V_{\mathrm{AS}}^2 + 0.2308V_{\mathrm{AS}} - 10.037$,
$X_\eta = 0.026$,
$X_\tau = 24.76$,
$Z_u = -8.66 \times 10^{-5}V_{\mathrm{AS}}^2 + 0.01249V_{\mathrm{AS}} - 0.5699$,
$Z_w = -0.08199V_{\mathrm{AS}} - 0.10375$,
$Z_\theta = -0.001358V_{\mathrm{AS}}^2 + 0.1679V_{\mathrm{AS}} - 5.4519$,
$Z_q = 1.012V_{\mathrm{AS}} - 0.77$,
$Z_\eta = 0.0001159V_{\mathrm{AS}}^2 + 4.04 \times 10^{-5}V_{\mathrm{AS}} - 0.001415$,
$M_u = 0.2.6146V_{\mathrm{AS}}^2 - 0.003426V_{\mathrm{AS}} - 0.1425$,
$M_w = -0.02817V_{\mathrm{AS}} + 0.001135$,
$M_\theta = -0.0001475V_{\mathrm{AS}}^2 + 0.01824V_{\mathrm{AS}} - 0.5922$,
$M_q = 0.04614V_{\mathrm{AS}} - 0.08368$,
$M_\eta = 0.00021V_{\mathrm{AS}}^2 - 0.00015V_{\mathrm{AS}} + 0.005065$,
$M_\tau = -0.294$.

In the following section, the LPV model in (15) is going to be used to design the SVOs for the FDI method described in Section 4. However, for the simulation of the dynamics of the UAV, the nonlinear model in (14) will be used instead.

6. Simulation results

This section presents a series of simulations, performed in MATLAB, that illustrate the applicability of the SVOs in fault detection and isolation for a fixed-wing aircraft. The aircraft nonlinear model (14) for the longitudinal axis, presented in Section 5, is used. By resorting to the LPV model in (15), discretized with a sampling period of $T_s = 10$ ms, a fault detection filter as in Section 4 was synthesized in order to diagnose faults in the longitudinal dynamics of the aircraft. As previously mentioned, notice that the LPV model is used solely for the design of the SVOs, while the nonlinear model is used for simulation purposes. Both the exogenous disturbances, $\boldsymbol{d}(k)$, and the sensors noise, $\boldsymbol{n}(k)$, were generated by using band-limited white noise. Moreover, three fault isolation filters were also designed using the approach in Section 4, in order to isolate the following failures in the aircraft:

1. FDI #1: additive fault in the angle-of-attack, α, sensor;
2. FDI #2: loss-of-effectiveness of the elevator, η;
3. FDI #3: increased pitch moment derivative with respect to longitudinal speed (system dynamics fault).

The aforementioned system dynamics fault represents a severe change in the dynamics of the aircraft caused, for instance, by a damaged stabilizer.

The FDI architecture used is depicted in Fig. 7. In this case, the FDI filters were designed so that loss-of-effectiveness types of faults can be diagnosed. Thus, as explained in detail in Section 4, the FDI filters need not be reset whenever a fault is detected by the nominal filter.

Remark 3. In the following simulations, the faults were assumed to be happening separately. Having simultaneous faults requires FDI/SVO filters that are specifically designed for more than one fault. Therefore, all the possible combinations of faults have to be considered, if more than a single fault is assumed to be occurring at a time.

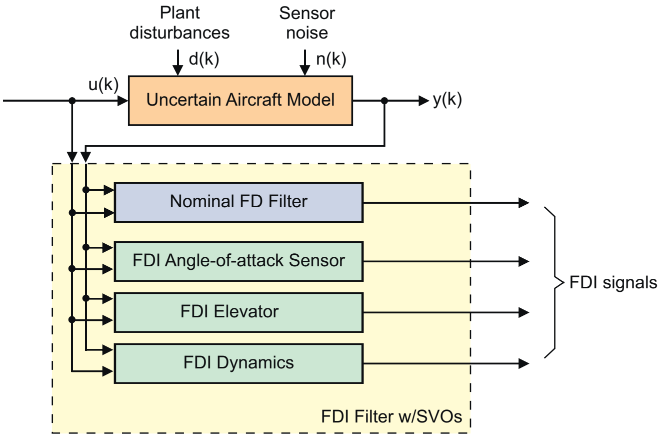

Fig. 7. FDI architecture using SVOs for the aircraft longitudinal model.

As described in Section 4, since three possible faults are considered, four SVOs are required for Architecture I. Indeed, set-valued state estimates are provided by

1. SVO #1, for the nominal (i.e., non-faulty) system;
2. SVO #2, for the system with a faulty angle-of-attack sensor;
3. SVO #3, for the system with a faulty elevator;
4. SVO #4, for the system with an uncertain pitch moment partial derivative with respect to u.

SVO #2 was designed by increasing the allowable α-sensor noise, so that additive sensor faults are compatible with the model of the system. For the third SVO, however, the fault cannot be handled by considering increased noise in the sensors. As described in Section 2, this is achieved instead by defining an uncertain input matrix, described by

$$\mathbf{B}(k) + \mathbf{B}_1(k)\boldsymbol{\Delta}(k).$$

Finally, the fourth SVO is designed by considering, for each k, an $\mathbf{A}(k)$ matrix with uncertain value of M_u—see (16).

An LQG controller was designed for the aircraft's linearized model, about the nominal airspeed of $V_{\mathrm{AS}}^{\mathrm{Trim}} = 37$ m/s and nominal angle-of-attach of $\alpha^{\mathrm{Trim}} = 0.01$ rad, and was connected to the loop.

Three different scenarios are going to be analyzed in the sequel.

6.1. Step-type faults

The first one consists in generating large step-type faults in the angle-of-attack sensor, in the elevator, and in the dynamics. Consider a step-type additive fault in the angle-of-attack, α, sensor. In particular, for $t \geq 0.5$ s, a constant signal with amplitude set at about 10% of the sensor range, which corresponds to 3°, is added to the noisy measurement of α, as depicted in Fig. 8.

For the elevator, a hard-type of fault was also considered. In particular, it is assumed that, for $t \geq 0.5$ s, the effectiveness of this actuator is suddenly decreased to 50% of its normal operating condition.

Finally, for the fault in the dynamics, it is considered that, for $t \geq 0.5$ s, the pitch moment partial derivative with respect to u, denoted by C_{M_u}, is increased by 30%.

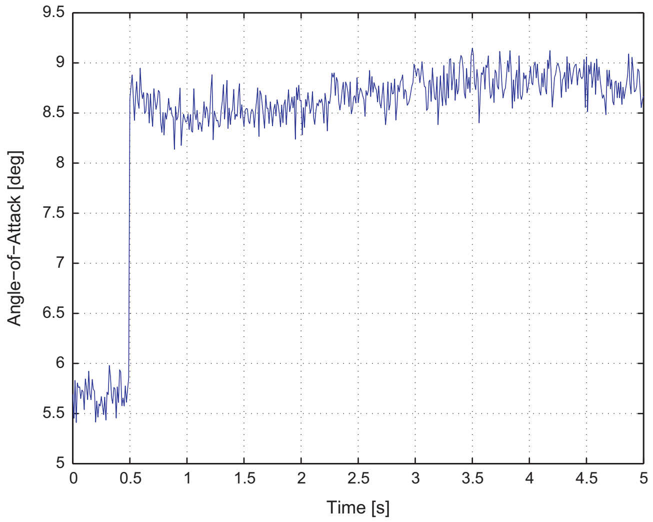

Fig. 8. Angle-of-attack measurement with an additive step-type fault in the sensor.

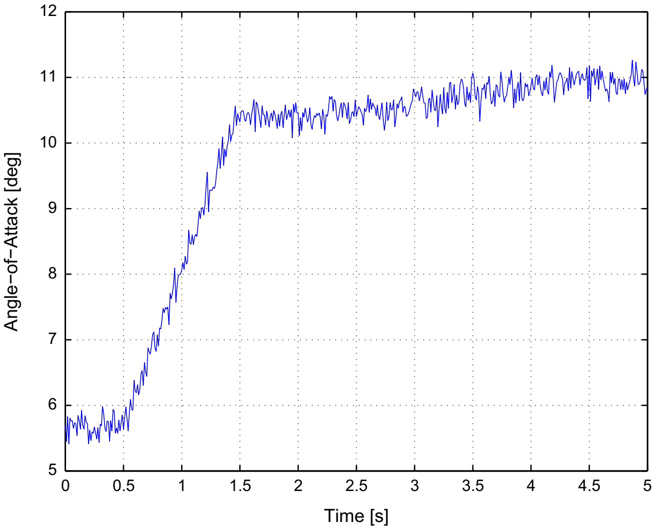

Fig. 9. Angle-of-attack measurement with an additive hard fault in the sensor.

6.2. *Hard faults*

It should be noticed, however, that step-type faults are, in general, "easier" to detect. Indeed, for the second scenario, faults in the angle-of-attack sensor, elevator, and dynamics, that can still be considered as hard faults, are analyzed, but assuming that they do not occur instantaneously, and thus represent more realistic failures. In particular, an additive fault in the angle-of-attack sensor is considered, as illustrated in Fig. 9, with $T_R = 1$ s and $A = 5^\circ$, where T_R and A denote the ramp duration and the amplitude of the fault, respectively. Regarding the elevator, it is assumed that the effectiveness of the surface linearly decreases in $T_R = 1$ s, from 100% to 50%. Finally, for the fault in the dynamics, the value of M_u is linearly increased, during $T_R = 1$ s, up to 130%.

6.3. *Soft faults*

As a final scenario, the so-called runaway or soft faults in the angle-of-attack sensor, in the elevator, and in the dynamics, are analyzed. For this case, let $T_R = 4$ s and $A = 5^\circ$, for the fault in the angle-of-attach sensor—see Fig. 10. Similarly to what was done in the previous scenarios, it was considered that the effectiveness of the elevator linearly decreases from 100% to 50% in $T_R = 4$ s, while the value of C_{M_u} is increased up to 130% of its nominal value, also in $T_R = 4$ s.

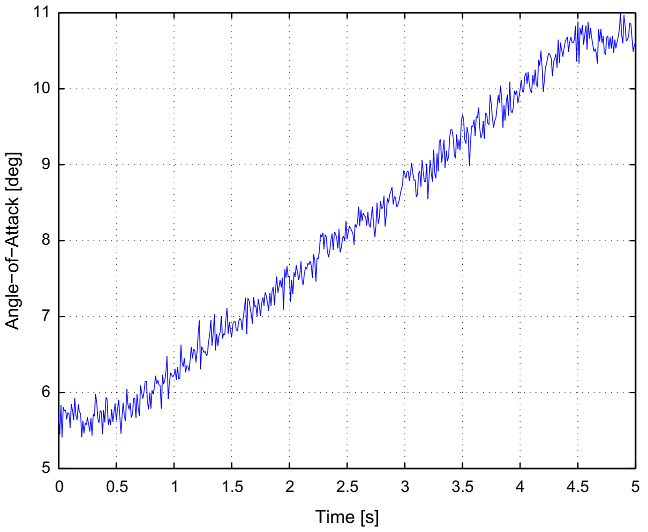

Fig. 10. Angle-of-attack (AOA) measurement with an additive soft (runaway) fault in the sensor.

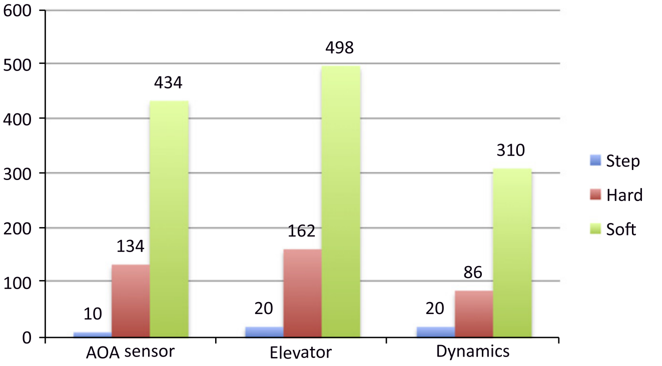

Fig. 11. Average fault detection times (in milliseconds), for several types of faults.

This type of faults is considerably harder to be detected and isolated, as the effects of such failures do not have an immediate impact on the measured data. Furthermore, to some extent, they may be indistinguishable from sensor noise or exogenous disturbances.

6.4. *Results—time-to-detect*

The results obtained by averaging 10 Monte-Carlo simulation runs for each fault considered are now described, in terms of the time required to detect the faults. Fig. 11 summarizes these results.

Notice that the sampling period is $T_s = 10$ ms, which means the step-type faults in the α-sensor were detected and isolated, for the present example, in a single measurement. This, however, only happened since this is a considerably severe and fast type of fault, which directly violates the dynamic model of the system, given the bounds on the measurement noise and exogenous disturbances. Nevertheless, if such bounds are violated in practice in the healthy system, then a false alarm may be issued by the algorithm. As stressed in Remark 2, the probability of having this type of false alarms is directly related to the likelihood of violating the bounds on the measurement noise and exogenous disturbances. It should also be stressed that the faults in the elevator and in the dynamics, even if severe, are only detected after 20 ms. This is due to the fact that these types of faults have to go through the dynamics of the system, before they can be measured by the sensors.

The smoother faults are, as expected, more difficult to be detected. In fact, since the proposed FDI method does not account for any a priori probability distribution regarding the exogenous disturbances and measurement noise, additive faults in the sensors with small amplitudes are not detected by this approach. Therefore, while the magnitude of the fault is comparable to the magnitude of the measurement noise, the non-faulty model of the plant is not invalidated. Nevertheless, the time required to detect each of these faults is below 0.5 s.

Remark 4. Each SVO requires between 5 and 200 ms per sampling period, to perform all the necessary computations, using an Intel Xeon CPU at 2.6 GHz, and depending on the horizon N in (3) Indeed, for larger values of N, the processing time required by each SVO may be above the sampling period, thus jeopardizing the implementability of the technique. Such an issue can be circumvented by (i) increasing the computational power, (ii) increasing the sampling period, or (iii) decreasing the horizon N. It should also be noticed that, due to their underlying structure, Architectures I and II can both benefit from standard parallel processing tools.

6.5. *Results—time-to-isolate*

The time required to isolate the faults is now analyzed. The results are summarized in Fig. 12.

A fault is isolated only when all but one SVO provides nonempty set-valued estimates of the state of the system. Hence, the time interval required to isolate a fault is typically much larger than the time required to detect it, and depends on the number of *admissible* faults, but also on whether a fault is distinguishable from another or not. In fact, the fault in the dynamics is, up to some extent, described by a severe failure on the elevator, as these two faults have similar consequences. Therefore, the time required to isolate these two faults is, in general, larger than the time required to isolate a fault in the angle-of-attack sensor.

Also notice that, in Samy et al. (2011), the time required to detect step-type faults in the α-sensor was 1 s. Therefore, the suggested approach is approximately 100 times faster, for the aforementioned assumptions in the measurement noise, intensity of the disturbances, and model uncertainty. However, the time required in Samy et al. (2011) to isolate soft faults in this sensor is approximately 1.86 s, while in the approach proposed in this paper nearly 3.5 s are required.

Remark 5. The detection and isolation of faults in a few milliseconds can pose several problems in terms of robustness of the method. For instance, outliers in the measurement noise, not satisfying the prescribed bounds, can lead to false positives. One possible heuristic to avoid such phenomena is to consider debounce for the detection and isolation of the faults. In particular, one can neglect a given number of previous measurements or use the *Global SVO* in Architecture II to reinitialize all the other SVOs a predefined number of times, before declaring a fault.

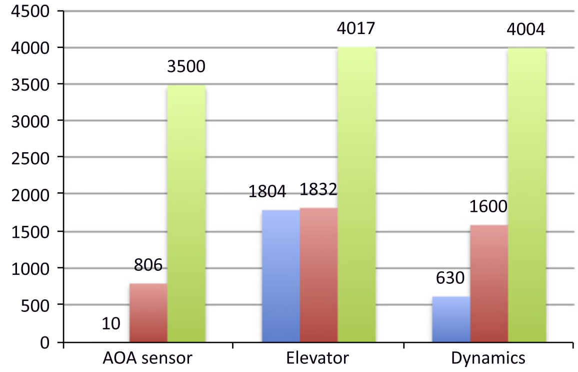

Fig. 12. Average fault isolation times (in milliseconds), for several types of faults.

7. Conclusions

This paper proposed and illustrated, with an example, a novel methodology that uses set-valued observers (SVOs) for fault detection and isolation (FDI). The suggested approach relies on the set-valued state estimates of the SVOs to validate or falsify the set of observations.

The SVOs were designed in such a way that model uncertainty and disturbances can be accounted for. Hence, *false alarms* due to such perturbations can be avoided. Moreover, unlike the typical FDI strategies available in the literature, one need not use residuals to decide whether a fault occurred. In such strategies, this decision naturally requires the definition of a (possibly time-varying) threshold, used to declare a fault whenever it is smaller than a given residual. The approach presented in this paper need not the definition of a threshold, and hence simplifies significantly the design of the FDI filters.

The applicability of the proposed FDI technique was demonstrated in simulation, using the full nonlinear model of the longitudinal dynamics of a fixed-wing aircraft, while the SVOs were developed using a linear parameter-varying (LPV) approximation of those dynamics. The simulation results show that the detection and isolation of the faults take, in general, only a few iterations. It was also noticed that, as expected, abrupt faults are easier to detect than smooth faults, using this approach. The results obtained indicate that this methodology holds considerable promise for practical applications.

The algorithms proposed, however, cannot be used, without performing any modification, to detect that a given fault has recovered. This goal may be achieved by restarting all the SVOs with the set-valued state estimate provided by the Global SVO, once the dynamics can again be described by the healthy model of the system.

Acknowledgements

This work was partially supported by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POS_ Conhecimento Program that includes FEDER funds, and by the PTDC/EEA-ACR/72853/2006 OBSERVFLY Project.

The authors would like to thank Professor Michael Athans and Professor Jeff S. Shamma for the many discussions on multiple-model approaches to system identification and control. The authors would also like to acknowledge the reviewers for their insightful comments and suggestions.

References

Alwi, H., & Edwards, C. (2008). Fault detection and fault-tolerant control of a civil aircraft using a sliding-mode-based scheme. *IEEE Transactions on Control Systems Technology*, *16*(3), 499–510.

Besançon, G. (2003). High-gain observation with disturbance attenuation and application to robust fault detection. *Automatica*, *39*, 1095–1102.

Bianchi, F., & Sánchez-Peña, R. (2010). Robust identification/invalidation in an LPV framework. *International Journal of Robust and Nonlinear Control*, *20*, 301–312.

Blanke, M., Izadi-Zamanabadi, R., Bogh, S., & Lunau, C. (1997). Fault tolerant control systems—a holistic view. *Control Engineering Practice*, *5*(5), 693–702.

Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., & Schröder, J. (2006). *Diagnosis and fault-tolerant control* (2nd ed.). Springer.

Blanke, M., Staroswiecki, M., & Wu, N. (2001). Concepts and methods in fault-tolerant control. In *Proceedings of the American control conference*, Arlington, VA, USA, June.

Bokor, J., & Balas, G. (2004). Detection filter design for LPV systems—a geometric approach. *Automatica*, *40*, 511–518.

Chen, J. (2010). *Robust linear parameter varying control of an unmanned aerial vehicle*. Ph.D. Thesis, University of Leicester, UK.

Collins, E. G. J., & Tinglun, S. (2001). Robust l_1 estimation using the Popov–Tsypkin multiplier with application to robust fault detection. *International Journal Control*, *74*(3), 303–313.

DiStefano, J., III (1977). On the relationships between structural identifiability and the controllability, observability, properties. *IEEE Transactions on Automatic Control*, *22*, 652.

DiStefano, J., III, & Cobelli, C. (1980). On parameter and structural identifiability: *Nonunique observability/reconstructibility for identifiable system, other ambiguities and new definitions*. *IEEE Transactions on Automatic Control*, *25*(4), 830–833.

Ducard, G. (2009). Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles. *Advances in industrial control*. Springer.

Edelmayer, A., Bokor, J., & Keviczky, L. (1994). An $\mathcal{H}_\infty$ approach to robust detection of failures in dynamical systems. In *Proceedings of the 33rd conference on decision and control* (Vol. 3, pp. 3037–3039), Lake Buena Vista, FL, USA.

Esteban, A. M. (2004). *Aircraft applications of fault detection and isolation techniques*. Ph.D. Thesis, University of Minnesota.

Frank, P., & Ding, X. (1994). Frequency domain approach to optimally robust residual generation. *Automatica*, *30*(5), 789–804.

Frank, P., & Ding, X. (1997). Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, *7*(6), 403–424.

Grewal, M., & Glover, K. (1976). Identifiability of linear and nonlinear dynamical systems. *IEEE Transactions on Automatic Control*, *21*(6), 833–837.

Hwang, I., Kim, S., Kim, Y., & Eng Seah, C. (2010). A survey of fault detection, isolation and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, *18*(3), 636–653.

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods—an introduction. *Control Engineering Practice*, *5*(5), 639–652.

Johansson, A. (2009). A sufficient condition for invalidation of linear state-space systems with uncertain time-varying parameters. *IEEE Transactions on Automatic Control*, *54*(12), 2915–2920.

Keerthi, S., & Gilbert, E. (1987). Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints. *IEEE Transactions on Automatic Control*, *32*(5), 432–435.

Longhi, S., & Moteriù, A. (2009). Fault detection for linear periodic systems using a geometric approach. *IEEE Transactions on Automatic Control*, *54*(7), 1637–1643.

Lou, H., & Si, P. (2009). The distinguishability of linear control systems. *Nonlinear Analysis: Hybrid Systems*, *3*, 21–38.

Mangoubi, R., Appleby, B., Verghese, G. & Vander Velde, W. (1995). A robust failure detection and isolation algorithm. In *Proceedings of the 34th conference on decision and control* (Vol. 3, pp. 2377–2382), New Orleans, USA.

Marcos, A., Ganguli, S., & Balas, G. (2005). An application of $\mathcal{H}_\infty$ fault detection and isolation to a transport aircraft. *Control Engineering Practice*, *13*, 105–119.

Massoumnia, M.-A. (1986a). *A geometric approach to failure detection and identification in linear systems*. Ph.D. Thesis, Massachusetts Institute of Technology.

Massoumnia, M. A. (1986b). Geometric approach to the synthesis of failure detection filters. *IEEE Transactions on Automatic Control*, *31*(9), 839–846.

Mattone, R., & De Luca, A. (2006). Relaxed fault detection and isolation: *An application to a nonlinear case study*. *Automatica*, *42*, 109–116.

Meskin, N., & Khorasani, K. (2009). Fault detection and isolation of discrete-time Markovian jump linear systems with application to a network of multi-agent systems having imperfect communication channels. *Automatica*, *45*, 2032–2040.

Milanese, M., & Vicino, A. (1991). Optimal estimation theory for dynamic systems with set membership uncertainty: *An overview*. *Automatica*, *27*(6), 997–1009.

Narasimhan, S., Vachhani, P., & Rengaswamy, R. (2008). New nonlinear residual feedback observer for fault diagnosis in nonlinear systems. *Automatica*, *44*, 2222–2229.

Niemann, H., & Stoustrup, J. (2001). Fault diagnosis for non-minimum phase systems using H_∞ optimization. In *Proceedings of the American control conference*, Arlington, VA, USA, June.

Patton, R., & Chen, J. (1997). Observer-based fault detection and isolation: *Robustness and applications*. *Control Engineering Practice*, *5*(5), 671–682.

Poolla, K., Khargonekar, P., Tikku, A., Krause, J., & Nagpal, K. (1994). A time-domain approach to model validation. *IEEE Transactions on Automatic Control*, *39*(5), 951–959.

Puig, V., Quevedo, J., Escobet, T., Nejjari, F., & de las Heras, S. (2008). Passive robust fault detection of dynamic processes using interval models. *IEEE Transactions on Control Systems Technology*, *16*(5), 1083–1089.

Rosa, P., & Silvestre, C. (2011). On the distinguishability of discrete linear time-invariant dynamic systems. In *Proceedings of the 50th IEEE conference on decision and control*, December.

Rosa, P., Silvestre, C., & Athans, M. (2011). Model falsification of LPV systems using set-valued observers. In *Proceedings of the 18th IFAC World Congress*, August–September.

Rosa, P., Silvestre, C., Shamma, J., & Athans, M. (2009). Multiple-model adaptive control with set-valued observers. In *Proceedings of the 48th IEEE conference on decision and control*, December.

Rosa, P., Silvestre, C., Shamma, J., & Athans, M. (2010). Fault detection and isolation of LTVs systems using set-valued observers. In *Proceedings of the 49th IEEE conference on decision and control*, December.

Samy, I., Postlethwaite, I., & Gu, D.-W. (2011). Survey and application of sensor fault detection and isolation schemes. *Control Engineering Practice*, *19*(7), 658–674.

Schweppe, F. (1968). Recursive state estimation: *Unknown but bounded errors and system inputs*. *IEEE Transactions on Automatic Control*, *13*(February (1)), 22–28.

Schweppe, F. (1973). *Uncertain dynamic systems*. USA: Prentice-Hall.

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

Stevens, B., & Lewis, F. (2003). *Aircraft control and simulation*. Wiley-Interscience.

Walter, E., Lecourtier, Y., & Happel, J. (1984). On the structural output distinguishability of parametric models, and its relations with structural identifiability. *IEEE Transactions on Automatic Control*, *29*(1), 56–57.

Wang, D., Wang, W., & Shi, P. (2009). Robust fault detection for switched linear systems with state delays. *IEEE Transactions on Automatic Control*, *39*(3), 800–805.

Willsky, A. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica*, *12*, 601–611.

Witsenhausen, H. (1968). Sets of possible states of linear systems given perturbed observations. *IEEE Transactions on Automatic Control*, *13*(5), 556–558.