

FAULT-TOLERANT CONTROL STRATEGY FOR ACTUATOR FAULTS USING LPV TECHNIQUES: APPLICATION TO A TWO DEGREE OF FREEDOM HELICOPTER

SAÚL MONTES DE OCA *, VICENÇ PUIG *, MARCIN WITCZAK **, ŁUKASZ DZIEKAN **

* Advanced Control Systems Group (SAC), Institute of Robotics and Industrial Informatics (IRI-CSIC)
Technical University of Catalunya (UPC), Pau Gargallo, 5, 08028 Barcelona, Spain
e-mail: {saul.montes.de.oca, vicenc.puig}@upc.edu

**Institute of Control and Computation Engineering
University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland
e-mail: {M.Witczak, L.Dziekan}@issi.uz.zgora.pl

In this paper, a Fault Tolerant Control (FTC) strategy for Linear Parameter Varying (LPV) systems that can be used in the case of actuator faults is proposed. The idea of this FTC method is to adapt the faulty plant instead of adapting the controller to the faulty plant. This approach can be seen as a kind of virtual actuator. An integrated FTC design procedure for the fault identification and fault-tolerant control schemes using LPV techniques is provided as well. Fault identification is based on the use of an *Unknown Input Observer* (UIO). The FTC controller is implemented as a state feedback controller and designed using polytopic LPV techniques and Linear Matrix Inequality (LMI) regions in such a way as to guarantee the closed-loop behavior in terms of several LMI constraints. To assess the performance of the proposed approach, a two degree of freedom helicopter is used.

Keywords: fault-tolerant control, linear parameter varying, virtual actuator, linear matrix inequality.

1. Introduction

Fault Tolerant Control (FTC) is one of the lines of research that have received a lot of interest in the last decades. According to Blanke *et al.* (1997), FTC allows maintaining the current performance close to desirable one and preserve stability conditions in the presence of component and/or instrument faults. The accommodation capability of a control system depends on many factors such as the severity of the fault, the robustness of the nominal system and mechanisms that introduce redundancy in sensors and/or actuators. A bibliographical review on the FTC approach can be found in the works of Patton (1997) as well as Zhang and Jiang (2008).

From the point of view of FTC strategies, the literature considers two main groups of techniques: *active* and *passive* ones. *Passive FTC techniques* are control laws that take into account fault appearance as a system perturbation. Thus, within certain margins, the control law has inherent fault tolerant capabilities, allowing the system to cope with the fault presence. Chen *et al.* (1998),

Liang *et al.* (2000) and Qu *et al.* (2003), among many others, provide complete descriptions of passive FTC techniques. On the other hand, *active FTC techniques* consist in adapting the control law using the information given by the FDI (Fault Detection and Isolation) block (Blanke *et al.*, 2006). With this information, some automatic adjustments are done trying to reach the control objectives. (See the works of Zhang and Jiang (2008) and Blanke *et al.* (2006) for a recent review of active FTC.) The whole active FTC scheme can be expressed using the three-layer architecture for FTC systems proposed by Blanke *et al.* (2006), where the first layer corresponds to the control loop, the second layer corresponds to the fault diagnosis and accommodation modules, while the third layer is the supervision system.

Active FTC strategies make it possible to handle bigger faults than passive ones can. Some research works following this approach include those by Maki *et al.* (2004), Rodrigues *et al.* (2005) or Zhang *et al.* (2005). Active FTC is characterized by the use of an on-line FDI scheme (Rodrigues *et al.*, 2008) and an automatic control

reconfiguration mechanism. On the other hand, fault accommodation has been addressed by Patton (1997). Recently, virtual sensors and actuators have been proposed as a fault accommodation approach for linear systems (Lunze, 2006; Richter et al., 2007; Blanke et al., 2006).

For non-linear systems, the design of FTC mechanisms is rather complicated. The use of multiple linear models represents an attractive solution for dealing with the control of nonlinear systems (Leith and Leithead, 1999; Banerjee et al., 1995). A similar idea is based on the use of Takagi–Sugeno fuzzy models (Murray-Smith and Johansen, 1997). Witczak et al. (2007) propose an integrated FTC design procedure of fault identification and fault-tolerant control for Takagi–Sugeno fuzzy systems using Lyapunov theory and LMIs. An alternative attractive solution is to represent the non-linear system as an LPV one. The main advantage of LPV models is that they allow applying powerful linear design tools to complex non-linear models (Hallouzi et al., 2005). Various candidate LPV system modeling techniques in the fault-free case are presented by Henrion et al. (2005) as well as Wan and Kothare (2003). LPV theory has been mainly used for designing controllers for non-faulty systems, but only recently for active FTC. Rodrigues et al. (2007) developed a solution to handle FTC and polytopic LPV systems with an Static Output Feedback (SOF) design.

In this paper, an FTC strategy based on a kind of *virtual actuator* approach for non-linear systems that can be approximated by LPV models is proposed. This strategy is inspired by the results presented by Witczak et al. (2007) that address the same problem but for Takagi–Sugeno systems. However, Witczak et al. (2007) do not take into account the control design that guarantee, the desired closed-loop behavior. This paper deals with an integrated FTC design procedure of fault identification, the virtual actuator and the state feedback controller. *Fault identification* is based on the use of a UIO. This approach allows detecting, isolating and estimating additive actuator faults. Once the fault has been identified, the *FTC controller* is implemented as a state feedback control. The idea of this FTC method is to adapt the faulty plant, instead of adapting the controller to the faulty plant. That is, the faulty plant together with the virtual actuator, mimics the fault-free plant. Finally, this strategy is designed using polytopic LPV techniques and LMI regions (Chilali and Gahinet, 1996). This approach based on approximating the LPV system in a polytopic way guarantees the desired specifications defining a set of LMI constraints. To assess the validity of the proposed approach, a two degree of freedom helicopter is used (Fee, 1998).

The paper is organized as follows. Section 2 presents details regarding the proposed FTC Strategy. Section 3 describes the polytopic LPV model. In Section 4, an integrated design procedure for an observer and a state feedback controller is developed. Finally, Section 5 describes

the two degree of freedom helicopter which illustrates the performance of the proposed approach.

2. LPV fault tolerant control strategy

2.1. Problem set-up. Let us consider a discrete-time LPV system in the state-space form for the fault-free case:

$$x(k+1) = A(\vartheta_k)x(k) + B(\vartheta_k)u(k), \quad (1)$$

$$y(k) = C(\vartheta_k)x(k), \quad (2)$$

where $x(k) \in \mathbb{R}^{n_x}$, $y(k) \in \mathbb{R}^{n_y}$ and $u(k) \in \mathbb{R}^{n_u}$ represent the state vector, the output vector and the control input vector, respectively. The matrices have the dimensions $A(\vartheta_k) \in \mathbb{R}^{n_x \times n_x}$, $B(\vartheta_k) \in \mathbb{R}^{n_x \times n_u}$ and $C(\vartheta_k) \in \mathbb{R}^{n_y \times n_x}$. ϑ_k is the system vector of time-varying parameters of dimension n_ϑ that varies within a polytope Θ . ϑ_k changes with the operating point scheduled by some measured system variables p_k ($p_k := p(k)$) that can be estimated using some known function $\vartheta_k = f(p_k)$, known as a scheduling function.

The LPV system (1)–(2) describes a set of linear systems parameterized by a scheduling variable denoted by p_k that can be seen as the linear approximation of a non-linear system at a parameterized set of equilibrium points (Apkarian et al., 1995). Equations (1)–(2) should be perceived as a reference model, and hence it is assumed that its state is available while the strategy for determining the control law $u(k)$ will be provided in the subsequent part of the paper.

Now, let us consider the LPV system (1)–(2) including additive actuator faults:

$$x_f(k+1) = A(\vartheta_k)x_f(k) + B(\vartheta_k)u_f(k) + L(\vartheta_k)f(k), \quad (3)$$

$$y_f(k) = C(\vartheta_k)x_f(k), \quad (4)$$

where $x_f(k) \in \mathbb{R}^{n_x}$, $y_f(k) \in \mathbb{R}^{n_y}$, $u_f(k) \in \mathbb{R}^{n_u}$ and $f(k) \in \mathbb{R}^{n_s}$ represent the state vector, the output vector, the control input vector and the fault vector, respectively. $L(\vartheta_k)$ stands for the actuator fault distribution matrix which is assumed to be known. It should be clearly pointed out that Eqns. (3) and (4) describe the behaviour of the real plant.

The main objective of this paper is to develop an FTC control strategy based on a kind of virtual actuator approach which can be used for determining the system input $u_f(k)$ such that $x_f(k+1)$ converges asymptotically to $x(k+1)$ independently of the presence of the fault $f(k)$. The idea of the *virtual actuator*¹ is to adapt the faulty plant instead of adapting the controller to the faulty plant as shown in Fig.1. In this way, the faulty plant, together with the virtual actuator block, allows the controller to see the same plant as before the fault.

¹Virtual actuators were originally proposed by Lunze (2006), Richter et al. (2007) and Blanke et al. (2006).

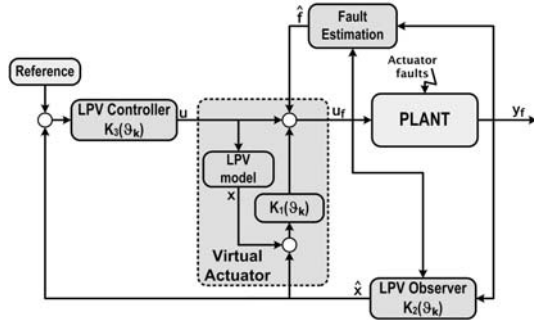


Fig. 1. Virtual actuator scheme.

In the proposed virtual actuator approach, the crucial idea is to determine the system input $u_f(k)$ as follows:

$$u_f(k) = -S\hat{f}(k) + K_1(\vartheta_k)(x(k) - \hat{x}_f(k)) + u(k), \quad (5)$$

where $\hat{f}(k)$ is the fault estimate. The purpose of the first factor $S\hat{f}(k)$ is to compensate the fault while the aim of the term $K_1(\vartheta_k)(\hat{x}(k) - \hat{x}_f(k))$ is to eliminate the control error. The term $u(k)$ is the control input provided by the controller. Since it is assumed that $x_f(k)$ is not available, $\hat{x}_f(k)$ need to be estimated.

2.2. Fault identification. Additive actuator faults in (3) can be identified by considering them to be an unknown input. This makes it possible to use the existing results on UIO theory (Hui and Žak, 2005; Witczak *et al.*, 2007) to identify the faults. The application of these results requires the following rank condition to be satisfied for all $\vartheta_k \in \Theta$ ²:

$$\text{rank}(C(\vartheta_k)L(\vartheta_k)) = \text{rank}(L(\vartheta_k)) = s. \quad (6)$$

This implies that it is possible to calculate the matrix $H(\vartheta_k)$ as follows:

$$\begin{aligned} H(\vartheta_k) &= (C(\vartheta_k)L(\vartheta_k))^+ \\ &= [(C(\vartheta_k)L(\vartheta_k))^T C(\vartheta_k)L(\vartheta_k)]^{-1} \\ &\quad \times (C(\vartheta_k)L(\vartheta_k))^T. \end{aligned} \quad (7)$$

Then, multiplying (4) by $H(\vartheta_k)$ and substituting (3), the fault estimate can be expressed as

$$\begin{aligned} f(k) &= H(\vartheta_k)(y_f(k+1) - C(\vartheta_k)A(\vartheta_k)x_f(k) \\ &\quad - C(\vartheta_k)B(\vartheta_k)u_f(k)). \end{aligned} \quad (8)$$

²This condition is not easy to guarantee unless matrices C and L are time invariant. However, in real life cases, checking if the rank condition is satisfied for a sparse grid of values of $\vartheta_k \in \Theta$ is usually sufficient.

Thus, if $\hat{x}_f(k)$ is used instead of $x_f(k)$, then the fault estimate is given as follows:

$$\begin{aligned} \hat{f}(k) &= H(\vartheta_k)(y_f(k+1) - C(\vartheta_k)A(\vartheta_k)\hat{x}_f(k) \\ &\quad - C(\vartheta_k)B(\vartheta_k)u_f(k)), \end{aligned} \quad (9)$$

and the associated fault estimation error is

$$f(k) - \hat{f}(k) = -H(\vartheta_k)C(\vartheta_k)A(\vartheta_k)(x_f(k) - \hat{x}_f(k)). \quad (10)$$

Unfortunately, the crucial problem with practical implementation of (9) is that it requires $y_f(k+1)$ and $u_f(k)$ to calculate $\hat{x}_f(k)$, and hence it cannot be directly used to obtain (5). To address this problem, it is assumed that there exists a diagonal matrix β_k such that $\hat{f}(k) = \beta_k \hat{f}(k-1)$ and hence the practical form of (5) boils down to³

$$u_f(k) = -S\beta_k \hat{f}(k-1) + K_1(\vartheta_k)(x(k) - \hat{x}_f(k)) + u(k), \quad (11)$$

where S satisfies $B(\vartheta_k)S = L(\vartheta_k)$.

2.3. Stabilization problem of a virtual actuator. By substituting (5) into (3), it can be shown that

$$\begin{aligned} x_f(k+1) &= A(\vartheta_k)x_f(k) - B(\vartheta_k)S\hat{f}(k) \\ &\quad + B(\vartheta_k)K_1(\vartheta_k)(e(k) + e_f(k)) \\ &\quad + B(\vartheta_k)u_k + L(\vartheta_k)f(k), \end{aligned} \quad (12)$$

where $e(k) = x(k) - x_f(k)$ stands for the tracking error while $e_f(k) = x_f(k) - \hat{x}_f(k)$ denotes the state estimation error.

Thus

$$\begin{aligned} x_f(k+1) &= A(\vartheta_k)x_f(k) + L(\vartheta_k)(f(k) - \hat{f}(k)) \\ &\quad + B(\vartheta_k)K_1(\vartheta_k)(e(k) + e_f(k)) \\ &\quad + B(\vartheta_k)u(k). \end{aligned} \quad (13)$$

Finally, substituting (10) into (13) yields

$$\begin{aligned} e(k+1) &= (A(\vartheta_k) - B(\vartheta_k)K_1(\vartheta_k))e(k) \\ &\quad + (L(\vartheta_k)H(\vartheta_k)C(\vartheta_k)A(\vartheta_k) \\ &\quad - B(\vartheta_k)K_1(\vartheta_k))e_f(k). \end{aligned} \quad (14)$$

³In most cases the matrix β_k should be equivalent to an identity matrix, i.e., it would simply mean a one time-step delay, which should have negligible effect on the outcome. In cases where the fault behaviour is nonlinear, it is possible to design the matrix β_k based on the previous changes of faults. In cases where a fault changes in a nonlinear fashion and one time-step delay is unacceptable, one could try to predict the nature of the faults by using, e.g., neural networks. In the work of Dziekan (2011) a similar problem was considered and it was found that a one time-step delay and a dynamic neural network trained for the fault trajectory gave a similar performance but with a substantial computational overhead. The best results were obtained for a ‘‘linear’’ predictor, but its performance was highly dependent on the selected parameters.

2.4. LPV observer. As already mentioned, the fault estimate (9) is obtained based on the state estimate $\hat{x}_f(k)$. This raises the necessity for an observer design. Consequently, by substituting (8) into (3), it is possible to show that

$$x_f(k+1) = \bar{A}(\vartheta_k)x_f(k) + \bar{B}(\vartheta_k)u_f(k) + \bar{L}(\vartheta_k)y_f(k+1), \quad (15)$$

where

$$\bar{A}(\vartheta_k) = (I - L(\vartheta_k))H(\vartheta_k)C(\vartheta_k)A(\vartheta_k),$$

$$\bar{B}(\vartheta_k) = (I - L(\vartheta_k))H(\vartheta_k)C(\vartheta_k)B(\vartheta_k)$$

and

$$\bar{L}(\vartheta_k) = L(\vartheta_k)H(\vartheta_k).$$

Thus, the observer structure, which can be perceived as an unknown input observer (see, e.g., Hui and Żak, 2005; Witczak et al., 2007) is given by

$$\begin{aligned} \hat{x}_f(k+1) &= \bar{A}(\vartheta_k)\hat{x}_f(k) + \bar{B}(\vartheta_k)u_f(k) + \bar{L}(\vartheta_k)y_f(k+1) \\ &\quad + K_2(\vartheta_k)(y_f(k) - C(\vartheta_k)\hat{x}_f(k)). \end{aligned} \quad (16)$$

Finally, the state estimation error can be written as

$$e_f(k+1) = (\bar{A}(\vartheta_k) - K_2(\vartheta_k)C(\vartheta_k))e_f(k). \quad (17)$$

2.5. LPV controller. The LPV system (3)–(4) is controlled by a state feedback control with tracking reference input as proposed by Franklin et al. (1997). The feedback control law is based on the classical state feedback:

$$u(k) = K_3(\vartheta_k)(x(k) - x_r(k)), \quad (18)$$

while an input reference gain $N_x(\vartheta_k)$ is added to the state feedback control law (18). The basic idea in determining the gain $N_x(\vartheta_k)$ is that it should transform the reference input $r(k)$ to a state reference $x_r(k)$ that corresponds to an equilibrium point for this $r(k)$:

$$N_x(\vartheta_k)r(k) = x_r(k). \quad (19)$$

Additionally, in order to remove the steady state error, a feed-forward control that is proportional to the reference input is added:

$$u_{ss}(k) = N_u(\vartheta_k)r(k). \quad (20)$$

Finally, taking into account the classical feedback control (18) and the gains $N_x(\vartheta_k)$ in (19) and $N_u(\vartheta_k)$ in (20), the control law can be expressed as follows:

$$\begin{aligned} u(k) &= u_{ss}(k) - K_3(\vartheta_k)(x(k) - x_r(k)), \\ &= N_u(\vartheta_k)r(k) - K_3(\vartheta_k)(x(k) - N_x(\vartheta_k)r(k)), \end{aligned} \quad (21)$$

where $N_x(\vartheta_k)$ and $N_u(\vartheta_k)$ can be designed extending the theory of Franklin et al. (1997) to LPV systems as

$$\begin{bmatrix} N_x(\vartheta_k) \\ N_u(\vartheta_k) \end{bmatrix} = \begin{bmatrix} A(\vartheta_k) - I & B(\vartheta_k) \\ C(\vartheta_k) & 0 \end{bmatrix}^+ \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad (22)$$

assuming that the following rank condition is satisfied: $\text{rank}(C(\vartheta_k)B(\vartheta_k)) = \text{rank}(B(\vartheta_k)) = n_u$. Thus, by substituting (21) into (1), it is possible to show that

$$\begin{aligned} x(k+1) &= (A(\vartheta_k) - B(\vartheta_k)K_3(\vartheta_k))x(k) \\ &\quad + B(\vartheta_k)u_{ss}(k) + B(\vartheta_k)K_3(\vartheta_k)x_r(k). \end{aligned} \quad (23)$$

2.6. Reconfiguration analysis. To analyze the reconfigured system, the *reconfiguration model* is considered. This augmented model includes the reference model (1), the tracking error (14) and the state estimation error (17) as follows⁴:

$$\begin{aligned} &\begin{bmatrix} x(k+1) \\ e(k+1) \\ e_f(k+1) \end{bmatrix} \\ &= \begin{bmatrix} A & 0 & 0 \\ 0 & A - BK_1 & LHCA - BK_1 \\ 0 & 0 & \bar{A} - K_2C \end{bmatrix} \begin{bmatrix} x(k) \\ e(k) \\ e_f(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u(k). \end{aligned} \quad (24)$$

By introducing the control law (21), the model of the reconfigured closed-loop behavior of the system (24) can be expressed as

$$\begin{aligned} &\begin{bmatrix} x(k+1) \\ e(k+1) \\ e_f(k+1) \end{bmatrix} \\ &= \begin{bmatrix} A - BK_3 & 0 & 0 \\ 0 & A - BK_1 & LHCA - BK_1 \\ 0 & 0 & \bar{A} - K_2C \end{bmatrix} \\ &\quad \times \begin{bmatrix} x(k) \\ e(k) \\ e_f(k) \end{bmatrix} + \begin{bmatrix} BK_3 \\ 0 \\ 0 \end{bmatrix} x_r(k) \\ &\quad + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u_{ss}(k). \end{aligned} \quad (25)$$

It can be easily observed from (25) that the *separation principle* can be applied following Ghersin and

⁴To simplify the equations, we assume that $A = A(\vartheta_k)$, $B = B(\vartheta_k)$, and so on.

Sanchez-Pena (2002), which suggests that an LPV system can be represented by a set of “frozen” LTI systems in the parameter variation set. Then, the set σ of eigenvalues of the closed-loop system (25) for each “frozen” LTI system consists of the set of eigenvalues of the nominal closed-loop system plus the tracking error and the state estimation error:

$$\begin{aligned} \sigma &= \sigma \{A - BK_3\} \cup \sigma \{A - BK_1\} \\ &\cup \sigma \{A - K_2C\}. \end{aligned} \quad (26)$$

Thus, the closed loop LPV controller, the LPV virtual sensor and the state LPV observer can be designed independently. Notice that

- the matrix $K_3(\vartheta_k)$ influences the behavior of $x(k)$ through $A(\vartheta_k) - B(\vartheta_k)K_3(\vartheta_k)$ (LPV controller);
- the matrix $K_1(\vartheta_k)$ affects the behavior of the tracking error $e(k)$ through $A(\vartheta_k) - B(\vartheta_k)K_1(\vartheta_k)$;
- the state estimation error $e_f(k)$ is affected by the matrix $K_2(\vartheta_k)$ through $A(\vartheta_k) - K_2(\vartheta_k)C(\vartheta_k)$ (the state LPV observer).

3. Polytopic approximation of an LPV system

3.1. Polytopic approximation. According to Apkarian *et al.* (1995), if the LPV system (1)–(2) can be approximated by a polytopic system, i.e., by a system whose state-space matrices range in a polytope of matrices, more computationally efficient results can be derived. The polytope of matrices is defined as the convex hull of a finite number of matrices N . Each polytope vertex corresponds to a particular value of the scheduling variable ϑ_k . In other words,

$$\begin{aligned} &\begin{pmatrix} A(\vartheta_k) & B(\vartheta_k) \\ C(\vartheta_k) & 0 \end{pmatrix} \\ &\in \text{Co} \left\{ \begin{pmatrix} A_j & B_j \\ C_j & 0 \end{pmatrix}, j = 1, \dots, N \right\} \\ &:= \sum_{j=1}^N \alpha_k^j(p_k) \begin{pmatrix} A_j & B_j \\ C_j & 0 \end{pmatrix}, \end{aligned} \quad (27)$$

where $\alpha_k^j(p_k)$, $j = 1, \dots, N$ are scheduling functions satisfying the constraints (a convex sum property) $\alpha_k^j(p_k) \geq 0$ and $\sum_{j=1}^N \alpha_k^j(p_k) = 1$.

3.2. Polytopic LPV system. The polytopic approximation of (1)–(2) can be expressed as follows

$$x(k+1) = \sum_{j=1}^N \alpha_k^j(p_k) [A_j x(k) + B_j u(k)], \quad (28)$$

$$y(k) = \sum_{j=1}^N \alpha_k^j(p_k) C_j x(k), \quad (29)$$

where $x(k) \in \mathbb{R}^{n_x}$, $u(k) \in \mathbb{R}^{n_u}$ and $y(k) \in \mathbb{R}^{n_y}$ represent the system state vector, the system control input and the output vector, respectively. Here $A_j \in \mathbb{R}^{n_x \times n_x}$, $B_j \in \mathbb{R}^{n_x \times n_u}$ and $C_j \in \mathbb{R}^{n_y \times n_x}$ are time-invariant matrices defined for the j -th model. The polytopic system is scheduled through functions designed as $\alpha_k^j(p_k)$, $\forall j \in [1, \dots, N]$, which lie in a convex set

$$\begin{aligned} \Omega &= \left\{ \alpha_k^j(p_k) \in \mathbb{R}^N, \right. \\ &\alpha_k(p_k) = [\alpha_k^1(p_k), \dots, \alpha_k^N(p_k)]^T, \\ &\left. \alpha_k^j(p_k) \geq 0, \forall j, \sum_{j=1}^N \alpha_k^j(p_k) = 1 \right\}. \end{aligned} \quad (30)$$

There are several ways of implementing (27) depending on how the functions $\alpha^j(p_k)$ are defined. Here, the function $\alpha^j(p_k)$ is defined via a barycentric combination of vertexes as suggested by Biannic (1996).

In the case of actuator faults, the polytopic LPV system (3)–(4) can be expressed as follows

$$\begin{aligned} x_f(k+1) &= \sum_{j=1}^N \alpha_k^j(p_k) [A_j x_f(k) + B_j u_f(k) \\ &+ L_j f(k)], \end{aligned} \quad (31)$$

$$y_f(k) = \sum_{j=1}^N \alpha_k^j(p_k) C_j x_f(k), \quad (32)$$

where $x_f(k) \in \mathbb{R}^{n_x}$, $u_f(k) \in \mathbb{R}^{n_u}$, $y_f(k) \in \mathbb{R}^{n_y}$ and $f(k) \in \mathbb{R}^{n_u}$ represent the system state vector, the system control input, the output vector and the actuator fault, respectively.

According to Apkarian *et al.* (1995), to design a state feedback control for the polytopic LPV system (31)–(32), matrices B , L and C should be constant. However, if this is not the case, this difficulty could be overcome by including pre-filtering and post-filtering in the polytopic LPV system (31)–(32)⁵, which removes the parameter dependency of matrices B , L and C as follows:

$$\tilde{x}_f(k+1) = \sum_{j=1}^N \alpha_k^j(p_k) [\tilde{A}_j \tilde{x}_f(k) + \tilde{B} \tilde{u}_f(k) + \tilde{L} \tilde{f}(k)], \quad (34)$$

$$\tilde{y}_f(k) = \tilde{C} \tilde{x}_f(k), \quad (35)$$

⁵Specifically, a new control input \tilde{u} , a new fault \tilde{f} and a new measured output \tilde{y} are defined as follows:

$$\begin{aligned} u_u(k+1) &= A_u x_u(k) + B_u \tilde{u}(k), \\ u(k) &= C_u x_u(k), \\ x_l(k+1) &= A_l x_l(k) + B_l \tilde{f}(k), \\ f(k) &= C_l x_l(k), \\ x_y(k+1) &= A_y x_y(k) + B_y y(k), \\ \tilde{y}(k) &= C_y(k) x_y(k), \end{aligned}$$

where A_u , A_l and A_y are stable. The resulting LPV plant is described

where $\tilde{x}_f(k) = [x_f(k) \ x_u(k) \ x_l(k) \ x_y(k)]^T$, $\tilde{u}_f(k), \tilde{y}_f(k)$ and $\tilde{f}(k)$ represent the augmented system state vector, the system control input, the output vector and the actuator fault, respectively.

Using the polytopic approximation of the LPV system (31)–(32) after the pre/post filtering, the fault estimation (9) can be expressed taking into account the polytopic approximation (27) as follows:

$$\hat{\tilde{f}}(k) = \sum_{j=1}^N \alpha_k^j \tilde{H}_j (\tilde{y}_f(k+1) - \tilde{C} \tilde{A}_j \hat{\tilde{x}}_f(k) - \tilde{C} \tilde{B} \tilde{u}_f(k)), \quad (36)$$

where $\tilde{H} = (\tilde{C} \tilde{L})^+$.

To obtain the polytopic LPV controller, the control law (21) is substituted in the control strategy (11)

$$\tilde{u}_f(k) = \sum_{j=1}^N \alpha_k(p_k) \left[\tilde{u}_{ss}(k) + \tilde{K}_{3,j} (\tilde{x}_f(k) - \tilde{x}_r(k)) - \tilde{S} \tilde{\beta}_k \hat{\tilde{f}}(k-1) + \tilde{K}_{1,j} (\tilde{x}(k) - \hat{\tilde{x}}_f(k)) \right], \quad (37)$$

where the matrices \tilde{S} are defined satisfying the equality $\tilde{B} \tilde{S} = \tilde{L}$ (see Eqn. (11)).

Analogously, the LPV observer (16) used to estimate $\hat{\tilde{x}}(t)$ can be implemented as follows:

$$\begin{aligned} \hat{\tilde{x}}_f(k+1) &= \sum_{j=1}^N \alpha_k^j \left[\tilde{A}_j \hat{\tilde{x}}_f(k) + \tilde{B} \tilde{u}_f(k) + \tilde{L} \tilde{y}_f(k+1) + \tilde{K}_{2,j} (\tilde{y}_f(k) - \tilde{C} \hat{\tilde{x}}_f(k)) \right], \quad (38) \end{aligned}$$

where the matrix $\tilde{K}_{2,j}$ is the state observer gain for the j -th model $j = 1, \dots, N$.

4. Fault tolerant control design for LPV systems

This section presents a design procedure for the proposed FTC strategy using the polytopic LPV system (31)–(32)

by

$$\begin{aligned} \tilde{x}(k+1) &= \begin{bmatrix} A(\theta_k) & B(\theta_k)C_u & L(\theta_k)C_l & 0 \\ 0 & A_u & 0 & 0 \\ 0 & 0 & A_l & 0 \\ B_y C(\theta_k) & 0 & 0 & A_y \end{bmatrix} \tilde{x}(k) \\ &+ \begin{bmatrix} 0 \\ B_u \\ 0 \\ 0 \end{bmatrix} \tilde{u}(k) + \begin{bmatrix} 0 \\ 0 \\ B_l \\ 0 \end{bmatrix} \tilde{f}(k), \\ \tilde{y}(k) &= [0 \ 0 \ 0 \ C_y] \begin{bmatrix} x(k) \\ x_u(k) \\ x_l(k) \\ x_y(k) \end{bmatrix}, \\ \tilde{x}(k) &= [x(k) \ x_u(k) \ x_l(k) \ x_y(k)]^T. \quad (33) \end{aligned}$$

and the LMI pole placement technique, which allows locating the poles inside the unit circle using an LMI region (Chilali and Gahinet, 1996). This design implies selecting

- matrices $\tilde{K}_{1,j}$ and $\tilde{K}_{3,j}$ of (37) in order to guarantee closed-loop stability of the system,
- matrices $\tilde{K}_{2,j}$ (see (38)) in order to correctly estimate the faulty system state by using the LPV state observer.

Additionally, the following assumptions are required to apply existing results on LPV systems (see Apkarian et al., 1995)

- for $\tilde{K}_{1,j}$ and $\tilde{K}_{3,j}$ of (37) to exist, the pair $(\tilde{A}(\vartheta_k), \tilde{B})$ should be stabilizable for all $\vartheta_k \in \Theta$;
- for $\tilde{K}_{2,j}$ of (38) exist, the pair $(\tilde{A}(\vartheta_k), \tilde{C})$ should be detectable for all $\vartheta_k \in \Theta$.

Under these assumptions, it is possible to design the matrices $\tilde{K}_{1,j}$, $\tilde{K}_{2,j}$ and $\tilde{K}_{3,j}$ using the polytopic reconfigured closed-loop augmented system (25) as follows:

$$\begin{aligned} \begin{bmatrix} \tilde{x}(k+1) \\ \tilde{e}(k+1) \\ \tilde{e}_f(k+1) \end{bmatrix} &= \sum_{j=1}^N \alpha_k(p_k) A_{0,j} \begin{bmatrix} \tilde{x}(k) \\ \tilde{e}(k) \\ \tilde{e}_f(k) \end{bmatrix} \\ &+ \begin{bmatrix} \tilde{B} \tilde{K}_{3,j} \\ 0 \\ 0 \end{bmatrix} \tilde{x}_r(k) + \begin{bmatrix} \tilde{B} \\ 0 \\ 0 \end{bmatrix} \tilde{u}_{ss}(k), \quad (39) \end{aligned}$$

where

$$\begin{aligned} A_{0,j} &= \begin{bmatrix} \tilde{A}_j - \tilde{B} \tilde{K}_{3,j} & 0 & 0 \\ 0 & \tilde{A}_j - \tilde{B} \tilde{K}_{1,j} & \tilde{L} \tilde{H} \tilde{C} \tilde{A}_j - \tilde{B} \tilde{K}_{1,j} \\ 0 & 0 & \tilde{A} - \tilde{K}_{2,j} \tilde{C} \end{bmatrix}. \end{aligned}$$

According to Chilali and Gahinet (1996), an LMI region is any subset \mathcal{D} of the complex plane that can be defined as

$$\mathcal{D} = \{z \in \mathbb{C} : P + zM + \bar{z}M^T < 0\}, \quad (40)$$

where P and M are real matrices such that $P^T = P$. The matrix-valued function $f_{\mathcal{D}}(z) = P + zM + \bar{z}M^T$ is called the characteristic function. This LMI region $\mathcal{D}(-q, r)$ is characterized by a disk of radius r and center q such that the characteristic function is given by

$$f_{\mathcal{D}}(z) = \begin{bmatrix} -r & q+z \\ q+\bar{z} & -r \end{bmatrix} < 0. \quad (41)$$

These two scalars q and r are used to determine a specific region included in the unit circle where to place closed-loop system eigenvalues.

Using the LMI region (41) to locate the poles of the augmented system (39), the following set of LMIs should be solved for all the vertices models $j \in [1, \dots, N]$

$$\begin{pmatrix} -rX & qX + (A_{0,j}X)^T \\ qX + A_{0,j}X & -rX \end{pmatrix} < 0, \quad (42)$$

where $A_{0,j}$ is stable if and only if there exists a symmetric matrix such that $X = X^T > 0$.

It can be observed from the structure of $A_{0,j}$ in (39) that the eigenvalues of the matrix $A_{0,j}$ are the union of $\tilde{A}_j - \tilde{B}\tilde{K}_{1,j}$, $\tilde{A}_j^T - \tilde{C}^T\tilde{K}_{2,j}^T$ and $\tilde{A}_j - \tilde{B}\tilde{K}_{3,j}$. This clearly indicates that the design of the state feedback, observer and controller can be carried out independently (separation principle). Thus, the following inequalities can be derived:

$$\begin{bmatrix} -r_1X_1 & q_1X_1 + X_1^T(\tilde{A}_j^T - \tilde{K}_{1,j}^T\tilde{B}^T) \\ (q_1 + \tilde{A}_j - \tilde{B}\tilde{K}_{1,j})X_1 & -r_1X_1 \end{bmatrix} < 0, \quad (43)$$

$$\begin{bmatrix} -r_2X_2 & q_2X_2 + X_2^T(\tilde{A}_j - \tilde{K}_{2,j}\tilde{C}) \\ (q_2 + \tilde{A}_j^T - \tilde{C}^T\tilde{K}_{2,j}^T)X_2 & -r_2X_2 \end{bmatrix} < 0, \quad (44)$$

$$\begin{bmatrix} -r_3X_3 & q_3X_3 + X_3^T(\tilde{A}_j^T - \tilde{K}_{3,j}^T\tilde{B}^T) \\ (q_3 + \tilde{A}_j - \tilde{B}\tilde{K}_{3,j})X_3 & -r_3X_3 \end{bmatrix} < 0, \quad (45)$$

for $j = 1, \dots, N$.

We should note that the expressions (43)–(45) are Bilinear Matrix Inequalities (BMIs), which cannot be solved with LMI tools. However, by introducing the new matrices $W_{1,j} = \tilde{K}_{1,j}X_1$, $W_{2,j} = \tilde{K}_{2,j}^T X_2$ and $W_{3,j} = \tilde{K}_{3,j}X_3$ it is possible to transform them into the following LMIs:

$$\begin{bmatrix} -r_1X_1 & q_1X_{1,j} + X_1^T\tilde{A}_j^T - W_{1,j}^T\tilde{B}^T \\ (q_1 + \tilde{A}_j)X_1 - \tilde{B}W_{1,j} & -r_1X_1 \end{bmatrix} < 0, \quad (46)$$

$$\begin{bmatrix} -r_2X_2 & q_2X_{2,j} + X_2^T\tilde{A}_j - W_{2,j}^T\tilde{C} \\ (q_2 + \tilde{A}_j^T)X_2 - \tilde{C}^TW_{2,j} & -r_2X_2 \end{bmatrix} < 0, \quad (47)$$

$$\begin{bmatrix} -r_3X_3 & q_3X_{3,j} + X_3^T\tilde{A}_j - W_{3,j}^T\tilde{C} \\ (q_3 + \tilde{A}_j^T)X_3 - \tilde{C}^TW_{3,j} & -r_3X_3 \end{bmatrix} < 0 \quad (48)$$

for $j = 1, \dots, N$.

Thus, the design procedure boils down to solving the LMIs (46)–(48), and then determining $\tilde{K}_{1,j} = W_{1,j}(X_1)^{-1}$, $\tilde{K}_{2,j} = (W_{2,j}(X_2)^{-1})^T$ and $\tilde{K}_{3,j} = W_{3,j}(X_3)^{-1}$ for $j = 1, \dots, N$.

5. Application example: A twin-rotor MIMO system

5.1. Description of the twin-rotor multiple input multiple output system. The Twin-Rotor MIMO System (TRMS) is a laboratory setup developed by Feedback Instruments Limited for advanced control experiments. The system is perceived as a challenging control engineering problem due to its high non-linearity, cross-coupling between its two axes, and inaccessibility of some of its states through measurements. The TRMS mechanical unit has two rotors (the main and tail rotors) driven by DC motors placed on a beam together with a counterbalance whose arm with a weight at its end is fixed to the beam at the pivot (Fig.2). The TRMS can rotate freely both in the horizontal and vertical planes.

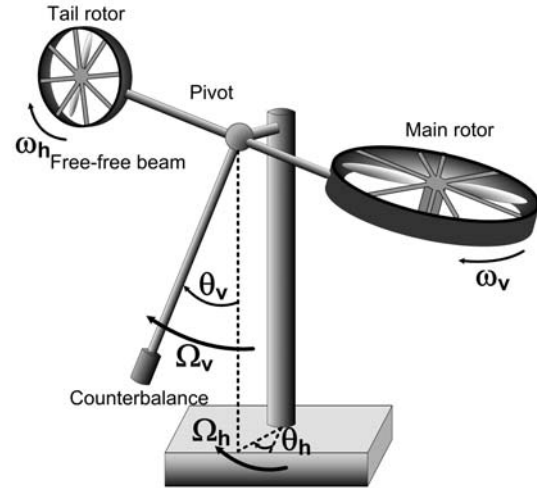


Fig. 2. Components of the twin rotor MIMO system.

The system input vector is $u = [u_t, u_m]^T$, where u_t is the input voltage of the tail motor and u_m is the input voltage of the main motor. The system states are $x = [\omega_h, \omega_v, \Omega_h, \Omega_v, \theta_h, \theta_v]^T$, where $\omega_{t/m}$ is the rotational velocity of the tail/main rotor, $\Omega_{h/v}$ is the angular velocity around the horizontal/vertical axis, and $\theta_{h/v}$ is the azimuth/pitch angle of the beam. Finally, the system measured outputs are $y = [\omega_v, \theta_h, \theta_v]^T$.

5.2. Polytopic LPV model of the TRMS. A polytopic LPV model is obtained by discretizing ($T = 0.05$ s) and linearizing the non-linear system around different operating points ($N = 4$ models). Thus, the polytopic LPV

representation (31)–(32) consists of the following matrices

$$\begin{aligned}
 A_1 &= \begin{bmatrix} 0.9812 & -0.0105 & 0.1847 \\ 0 & 0.9657 & 0 \\ 0 & 0 & 0.878 \\ 0 & 0.0152 & -0.0254 \\ 0 & 0.0004 & 0.1367 \\ 0.0495 & 0.0276 & 0.0047 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.9908 & -0.1718 & 0 \\ 0.0498 & 0.9957 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\
 A_2 &= \begin{bmatrix} 0.9814 & -0.0103 & 0.1841 \\ 0 & 0.9657 & 0 \\ 0 & 0 & 0.878 \\ 0 & 0.02 & -0.0254 \\ 0 & 0.0005 & 0.1367 \\ 0.0495 & 0.0274 & 0.0046 \\ 0 & 0.0004 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.9908 & -0.1718 & 0 \\ 0.0498 & 0.9957 & 0 \\ 0 & -0.001 & 1 \end{bmatrix}, \\
 A_3 &= \begin{bmatrix} 0.9818 & -0.0098 & 0.183 \\ 0 & 0.9657 & 0 \\ 0 & 0 & 0.878 \\ 0 & 0.0405 & -0.0254 \\ 0 & 0.001 & 0.1367 \\ 0.0495 & 0.0268 & 0.0045 \\ 0 & 0.0007 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.9908 & -0.1718 & 0 \\ 0.0498 & 0.9957 & 0 \\ -0.0001 & -0.002 & 1 \end{bmatrix}, \\
 A_4 &= \begin{bmatrix} 0.9826 & -0.009 & 0.1809 \\ 0 & 0.9657 & 0 \\ 0 & 0 & 0.878 \\ 0 & 0.0734 & -0.0254 \\ 0 & 0.0018 & 0.1367 \\ 0.0496 & 0.0256 & 0.0044 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.9908 & -0.1717 & 0 \\ 0.0498 & 0.9957 & 0 \\ -0.0001 & -0.003 & 1 \end{bmatrix}, \\
 B_1 &= \begin{bmatrix} 47.2 & -2.6 \\ 0.0 & 491.4 \\ 468.8 & 0.0 \\ -5.4 & 5.0 \\ 35.0 & 0.1 \\ 0.8 & 6.9 \end{bmatrix} \times 10^{-4}, \\
 B_2 &= \begin{bmatrix} 46.9 & -2.5 \\ 0.0 & 491.4 \\ 468.8 & 0.0 \\ -5.4 & 10.2 \\ 35.0 & 0.2 \\ 0.8 & 6.8 \end{bmatrix} \times 10^{-4}, \\
 B_3 &= \begin{bmatrix} 46.3 & -2.3 \\ 0.0 & 491.4 \\ 468.8 & 0.0 \\ -5.4 & 18.5 \\ 35.0 & 0.3 \\ 0.7 & 6.5 \end{bmatrix} \times 10^{-4}, \\
 B_4 &= \begin{bmatrix} 45.4 & -2.0 \\ 0.0 & 491.4 \\ 468.8 & 0.0 \\ -5.4 & 28.4 \\ 35.0 & 0.5 \\ 0.7 & 6.0 \end{bmatrix} \times 10^{-4}, \\
 C_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 896.24 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
 C_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 894.14 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
 C_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 879 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
 C_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 851.01 & 0 & 0 & 0 & 0 \end{bmatrix},
 \end{aligned}$$

where the scheduling variable is the azimuth angle of the beam $\theta_h(k)$.

5.3. FTC design. The virtual actuator $K_1(\vartheta_k)$ is obtained by solving the LMI (46) and locating the eigenvalues in a disk LMI region with center $q_1 = 0$ and radius $r_1 = 0.8$.

Analogously, the LPV observer $K_2(\vartheta_k)$ is obtained solving the LMI (47) placing the eigenvalues in the disk LMI region with center $q_2 = 0$ and radius $r_2 = 0.2$.

Finally, the LPV controller $K_3(\vartheta_k)$ is designed solving the LMI (48) considering a disk LMI region with center $q_3 = -0.8$ and radius $r_3 = 0.15$.

5.4. Fault scenarios. To assess the performance of the proposed approaches, two fault scenarios are considered. In both, the controller set-points are defined as follows:

$$\theta_h = \begin{cases} 0.0, & k < 1000, \\ 0.4, & 1000 \leq k < 5000, \\ 0.1, & k \geq 5000, \end{cases} \quad (49)$$

$$\theta_v = \begin{cases} 0.0, & k < 1000, \\ 0.06, & 1000 \leq k < 5000, \\ 0.0, & k \geq 5000. \end{cases} \quad (50)$$

This is to see the ability of the FTC controller to control the system and tolerate the faults in two operating points.

5.4.1. Fault scenario 1. In this scenario, a fault in the tail rotor $f_t(k)$ and the main rotor $f_m(k)$ is introduced as follows:

$$f_t(k) = \begin{cases} 0, & k < 3000, \\ -0.075, & 3000 \leq k < 4000, \\ 0, & k \geq 4000, \end{cases} \quad (51)$$

$$f_m(k) = \begin{cases} 0, & k < 6000, \\ 0.05, & 6000 \leq k < 8000, \\ 0, & k \geq 8000, \end{cases} \quad (52)$$

Figure 3 presents the control performance of the azimuth angle of the beam with and without the virtual actuator based FTC strategy. The system is stabilized with the virtual actuator approach in spite of the actuator fault, whereas without including such an FTC mechanism the angle of beam is not under control while the fault is present. Figure 4 shows the pitch angle of the beam. When using the virtual approach, it can be noticed, that the azimuth trajectory was not changed significantly after the fault occurrence. On the other hand, notice that, when the virtual actuator strategy is not used, the system is not able to track the reference. In this case, when the virtual actuator is not included, the controller is not able to tolerate the additive actuator fault.

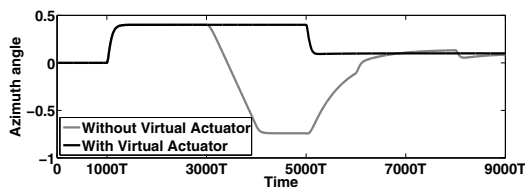


Fig. 3. Azimuth angle of the beam (horizontal position).

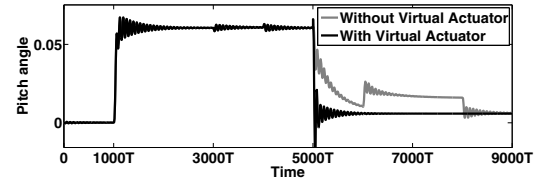


Fig. 4. Pitch angle of the beam (vertical position).

Figures 5 and 6 show the actuator faults and their estimations. Fault estimation presents a deviation of the nominal value due to changes in the reference. However, fault estimation achieves the real value in both the cases after the transient.

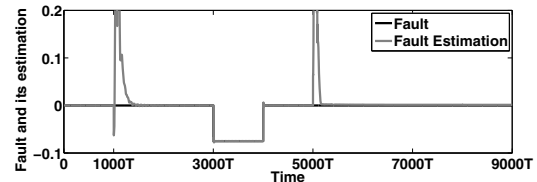


Fig. 5. Input voltage fault of the tail motor and its estimate.

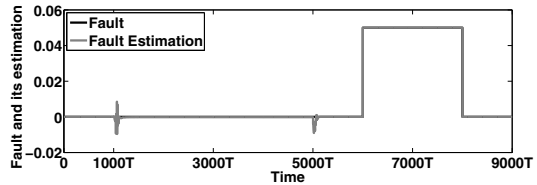


Fig. 6. Input voltage fault of the main motor and its estimate.

5.4.2. Fault scenario 2. In this scenario, a fault in the tail and the main rotor is introduced as follows:

$$f_h(k) = \begin{cases} 0, & k < 3000, \\ 0.05, & 3000 \leq k < 5000, \\ 0, & k \geq 5000, \end{cases} \quad (53)$$

$$f_v(k) = \begin{cases} 0, & k < 5000, \\ -0.05, & k \geq 5000. \end{cases} \quad (54)$$

Notice that the fault appears when the system adapts to the second operating point.

Figures 7 and 8 present the azimuth and the pitch angle of the beam, respectively, when the virtual actuator strategy is applied and, for comparison, when it is disabled. Notice that in this case, without the virtual actuator strategy, the system is not able to track the reference. Figures 9 and 10 present the input voltage applied to the main rotor and that the fault can be estimated with very high accuracy, except maybe in the transient.

6. Conclusions

In this paper, an active FTC strategy based on a kind of virtual actuator for non-linear systems that can be approxi-

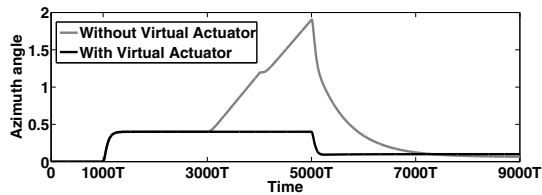


Fig. 7. Azimuth angle of the beam (horizontal position).

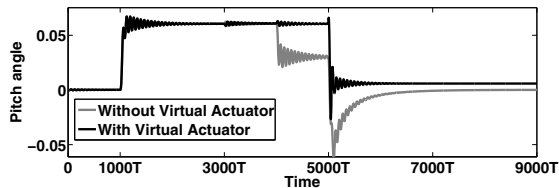


Fig. 8. Pitch angle of the beam (vertical position).

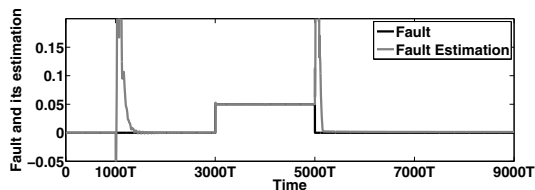


Fig. 9. Input voltage fault of the tail motor and its estimate.

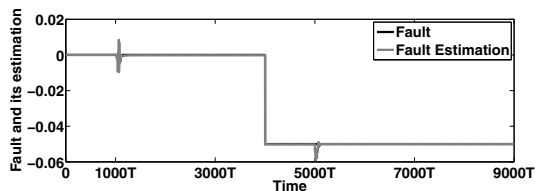


Fig. 10. Input voltage fault of the main motor and its estimate.

mated by LPV models has been presented. The key contribution of the proposed approach is an integrated FTC design procedure for fault identification, the virtual actuator and fault-tolerant control schemes using LPV techniques. Fault identification is based on the use of an UIO. The idea of the virtual actuator is to adapt the faulty plant instead of adapting the controller to the faulty plant. FTC strategy is designed through LMI pole placement. The proposed design of the FTC strategy places the eigenvalues of the closed-loop system in a predetermined LMI region inside the unit circle. The performance of the proposed approach has been satisfactorily assessed using a two degree of freedom helicopter.

Acknowledgment

This work has been partially funded by the grant CI-CYT DPI2008-01996 of the Spanish Ministry of Edu-

cation, by the Research Commission of Generalitat de Catalunya (SAC group ref. 2005SGR00537), by a grant from Consejo Nacional de Ciencia y Tecnología de México (CONACyT), and by the grant N N514 678440 of the Ministry of Science and Higher Education in Poland (within the science funds for the years 2011–2014).

References

- Apkarian, P., Gahinet, P. and Becker, G. (1995). Self-scheduled H_∞ control of linear parameter-varying systems: A design example, *Automatica* **31**(9): 1251–1261.
- Banerjee, A., Arkun, Y., Pearson, R. and Ogunnaiké, B. (1995). H_∞ control of nonlinear processes using multiple linear models, *Proceedings of the European Control Conference, Rome, Italy*, pp. 2671–2676.
- Biannic, J.M. (1996). *Commande Robuste des Systèmes à Paramètres Variables. Application en Aéronautique*, Ph.D. thesis, Study and Research Centre of Toulouse, DERA Department, Toulouse.
- Blanke, M., Izadi-Zamanabadi, R., Bogh, S.A. and Lunau, C.P. (1997). Fault-tolerant control systems—A holistic view, *Control Engineering Practice* **5**(5): 693–702.
- Blanke, M., Kinnaert, M., Lunze, J. and Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control*, Springer-Verlag, Berlin/Heidelberg.
- Chen, J., Patton, R.J. and Chen, Z. (1998). An LMI approach to fault-tolerant control of uncertain systems, *IEEE International Symposium on Intelligent Control, Gaithersburg, MD, USA*, Vol. 1, pp. 175–180.
- Chilali, M. and Gahinet, P. (1996). H_∞ design with pole placement constraints: An LMI approach, *IEEE Transactions on Automatic Control* **41**(3): 358–367.
- Dzianek, Ł. (2011). *Neuro-Fuzzy-Based Takagi–Sugeno Modelling in Fault-Tolerant Control*, Lecture Notes in Control and Computer Science, Vol. 16, University of Zielona Góra Press, Zielona Góra.
- Fee (1998). *Twin Rotor MIMO System Advanced Teaching Manual 1 (33-007-4M5)*.
- Franklin, G.F., Powell, J.D. and Workman, M.L. (1997). *Digital Control of Dynamic Systems*, 3rd Edn., Addison Wesley Longman, London.
- Ghersin, A.S. and Sanchez-Pena, R.S. (2002). LPV control of a 6-DOF vehicle, *IEEE Transactions on Control Systems Technology* **10**(6): 883–887.
- Hallouzi, R., Verdult, V., Babuska, R. and Verhaegen, M. (2005). Fault detection and identification of actuator faults using linear parameter varying models, *16th IFAC Triennial World Congress, Prague, Czech Republic*, pp. 119–124.
- Henrion, L.R.D., Bernussou, J. and Vary, F. (2005). LPV modeling of a turbofan engine, *Preprints of the 16th World Congress of the International Federation of Automatic Control, Prague, Czech Republic*, pp. 526–531.
- Hui, S. and Žak, S.H. (2005). Observer design for systems with unknown inputs, *International Journal of Applied Mathematics and Computer Science* **15**(4): 101–117.

- Leith, D. and Leithead, W. (1999). Survey of gain-scheduling analysis design, *International Journal of Control*, **73**(11): 1001–1025.
- Liang, Y., Liaw, D. and Lee, T. (2000). Reliable control of nonlinear systems, *IEEE Transactions on Automatic Control* **45**(4): 706–710.
- Lunze, J. (2006). Control reconfiguration after actuator failures: The generalised virtual actuator, *Proceedings of the 6th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS), Beijing, China*, pp. 1309–1314.
- Maki, M., Jiang, J. and Hagino, K. (2004). A stability guaranteed active fault-tolerant control system against actuator failures, *International Journal of Robust and Nonlinear Control* **14**(12): 1061–1077.
- Murray-Smith, R. and Johansen, T.A. (1997). *Multiple Model Approaches to Modelling and Control*, Taylor and Francis, London.
- Patton, R.J. (1997). Fault-tolerant control systems: The 1997 situation, *Proceedings of the IFAC Symposium: SAFEPROCESS'97, Hull, UK*, Vol. 2, pp. 1033–1055.
- Qu, Z., Ihlefeld, C. M., Yufang, J. and Saengdeejing, A. (2003). Robust fault-tolerant self-recovering control of nonlinear uncertain systems, *Automatica* **39**(10): 1763–1771.
- Richter, J.H., Schlage, T. and Lunze, J. (2007). Control reconfiguration of a thermofluid process by means of a virtual actuator, *IET Proceedings on Control Theory and Applications* **1**(6): 1606–1620.
- Rodrigues, M., Theilliol, D., Aberkane, S. and Sauter, D. (2007). Fault tolerant control design for polytopic LPV systems, *International Journal of Applied Mathematics and Computer Science* **17**(1): 27–37, DOI: 10.2478/v10006-007-0004-5.
- Rodrigues, M., Theilliol, D., Medina, M. A. and Sauter, D. (2008). A fault detection and isolation scheme for industrial systems based on multiple operating models, *Control Engineering Practice* **16**(2): 225–239.
- Rodrigues, M., Theilliol, D. and Sauter, D. (2005). Design of an active fault tolerant control and polytopic unknown input observer for systems described by a multi-model representation, *44th IEEE Conference on Decision and Control/European Control Conference ECC, Sevilla, Spain*, pp. 3816–3820.
- Wan, Z. and Kothare, M.V. (2003). Efficient scheduled stabilizing output feedback model predictive control for constrained nonlinear systems, *Proceedings of the American Control Conference, Denver, CO, USA*, Vol. 1 (4–6), pp. 489–494.
- Witczak, M., Dziekan, L., Puig, V. and Korbicz, J. (2007). An integrated design strategy for fault identification and fault-tolerant control for Takagi–Sugeno fuzzy systems, *Preprints of the 17th World Congress of the International Federation of Automatic Control, Seoul, Korea*, pp. 7387–7392.
- Zhang, Y. and Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems, *Annual Reviews in Control* **32**(2): 229–252.

- Zhang, Y.M., Jiang, J., Yang, Z. and Hussain, D.M.A. (2005). Managing performance degradation in fault tolerant control systems, *Preprints of the 16th IFAC Triennial World Congress, Prague, Czech Republic*, pp. 424–429.



Saúl Montes de Oca received the Ph.D. and Master's degree in automatic control, robotics and computer vision in 2011 from the Technical University of Catalonia (UPC), Barcelona, Spain, an M.Sc. in automation in 2006 and a B.S. in mechanical engineering in 2003 from Instituto Tecnológico y de Estudios Superiores de Monterrey, México. He is currently a professor of Universidad Autónoma de Guadalajara. His main research interests are fault detection and isolation, and fault-tolerant control of dynamic systems. He has published several papers in international conference proceedings.



Vicenç Puig received the Ph.D. degree in control engineering in 1999 and the telecommunications engineering degree in 1993, both from the Technical University of Catalonia (UPC), Barcelona, Spain. He is currently an associate professor of automatic control and the leader of the Advanced Control Systems (SAC) research group at UPC. His main research interests are fault detection and isolation of fault-tolerant control of dynamic systems. He has been involved in several European projects and networks and has published several papers in international conference proceedings and scientific journals.



Marcin Witczak received the M.Sc. degree in electrical engineering from the University of Zielona Góra (Poland), the Ph.D. degree in automatic control and robotics from the Wrocław University of Technology (Poland), and the D.Sc. degree in electrical engineering from the University of Zielona Góra in 1998, 2002 and 2007, respectively. He has been a professor of automatic control and robotics at the Institute of Control and Computation Engineering, University of Zielona Góra, since 2008. His current research interests include computational intelligence, Fault Detection and Isolation (FDI), Fault-Tolerant Control (FTC), and control theory. Marcin Witczak has published more than 100 papers in international journals and conference proceedings. He is an author of two monographs and nine book chapters, including *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems* (Springer).



Łukasz Dziekan received the M.Sc. and the Ph.D. degree in computer science from the University of Zielona Góra (Poland) in 2006 and 2011, respectively. He has been an assistant professor of automatic control and robotics at the Institute of Control and Computation Engineering, University of Zielona Góra, since 2011. His current research interests include computational intelligence, fuzzy logic, Fault Detection and Diagnosis (FDD), and Fault-Tolerant Control (FTC).

Received: 11 January 2011

Revised: 5 July 2011

Re-revised: 8 December 2011